

Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) **EP 1 294 127 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
19.03.2003 Bulletin 2003/12

(51) Int Cl.7: **H04L 12/24, H04L 12/26,
H04L 12/58**

(21) Application number: **02020531.6**

(22) Date of filing: **16.09.2002**

(84) Designated Contracting States:
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
IE IT LI LU MC NL PT SE SK TR**
Designated Extension States:
AL LT LV MK RO SI

(72) Inventors:
• **Motoyama, Tetsuro**
San Jose, CA 95131-1817 (US)
• **Fong, Avery**
San Jose, CA 95131-1817 (US)

(30) Priority: **17.09.2001 US 953359**

(74) Representative: **Schwabe - Sandmair - Marx**
Stuntzstrasse 16
81677 München (DE)

(71) Applicant: **Ricoh Company, Ltd.**
Tokyo 143-8555 (JP)

(54) **System and method for sending device configuration information to a monitor using e-mail**

(57) In a monitoring system for networked devices, a system, method, and computer program product for obtaining a globally-unique identifier for a network device, and sending the same to a central monitor responsible for monitoring devices on multiple networks. A remote monitoring workstation monitors devices connected to a network using IP-based SNMP commands and stores the device monitoring information in a database. The remote monitoring workstation queries the database to obtain an IP address and the collected monitoring information for a device, uses the IP address to request a globally-unique identifier (e.g., a MAC address) from a device using SNMP commands, then sends the monitoring information including the globally-unique device identifier to a central monitoring workstation via an e-mail message.

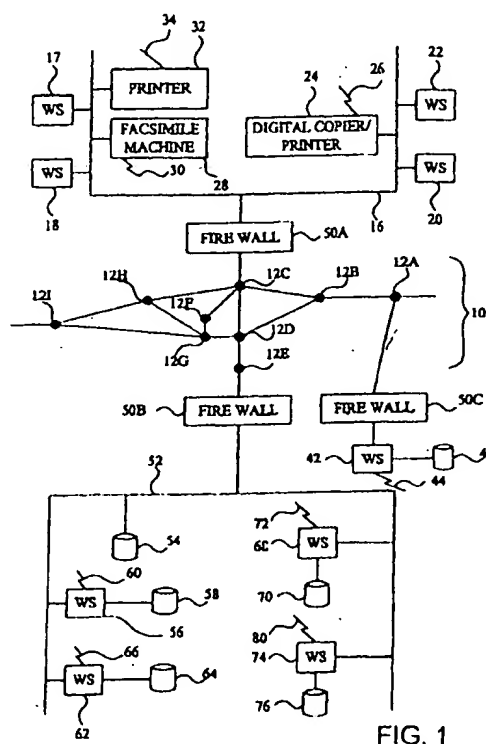


FIG. 1

EP 1 294 127 A1

Description

CROSS-REFERENCE TO RELATED APPLICATIONS

5 [0001] The present application is related to U.S. Patent Application 09/190,460, filed November 13, 1998, entitled "Method and System for Translating Documents Using Different Translation Resources for Different Portions of the Documents," which is a continuation of U.S. Patent Application 08/654,207, filed May 28, 1996, entitled "Method and System for Translating Documents Using Different Translation Resources for Different Portions of the Documents," now U.S.P. 5,848,386; U.S. Patent Application 08/997,705, filed December 23, 1997, entitled "Object-oriented System and Computer Program Product for Mapping Structured Information to Different Structured Information," now U.S.P. 6,085,196; U.S. Patent Application 08/997,705, filed December 23, 1997, entitled "Method and Apparatus for Providing a Graphical User Interface for Creating and Editing a Mapping of a First Structural Description to a Second Structural Description"; US Patent Application 09/756,120, filed January 9, 2001, entitled "Method and System of Remote Support Of Device Using Email"; U.S. Patent Application 09/668,162, filed September 25, 2000, entitled "Method and System of Data collection and Mapping From a Remote Position Reporting Device"; U.S. Patent Application 09/575,710, filed July 25, 2000, entitled "Method and System of Remote Diagnostic and Information Collection and Service System"; U.S. Patent Application 09/575,702, filed July 12, 2000, entitled "Method and System of Remote Position Report Device"; U.S. Patent Application 09/453,934, filed May 17, 2000, entitled "Method and System of Remote Diagnostic, Control and Information Collection Using a Dynamic Linked Library for Multiple Formats and Multiple Protocols"; U.S. Patent Application 09/453,935, filed May 17, 2000, entitled "Method and System of Remote Diagnostic, Control and Information Collection Using a Dynamic Linked Library of Multiple Formats and Multiple Protocols With Intelligent Protocol Processor"; U.S. Patent Application 09/453,937, filed May 17, 2000, entitled "Method and System of Remote Diagnostic, Control and Information Collection Using a Dynamic Linked Library of Multiple Formats and Multiple Protocols With Restriction on Protocol"; U.S. Patent Application 09/453,936, filed May 17, 2000, entitled "Method and System of Remote Diagnostic, Control and Information Collection Using a Dynamic Linked Library of Multiple Formats and Multiple Protocols with Intelligent Formatter"; U.S. Patent Application 09/542,284, filed April 4, 2000, entitled "System and Method to Display Various Messages While Performing the Tasks or While Idling"; U.S. Patent Application 09/520,368, filed on March 7, 2000, entitled "Method and System for Updating the Device Driver of a Business Office Appliance"; U.S. Patent Application 09/453,877, filed February 4, 2000, entitled "Method and System for Maintaining a Business Office Appliance through Log Files"; U.S. Patent Application 09/440,692, filed November 16, 1999, entitled "Method and System to Monitor the Application Usage and Send Back the Information Using Connection and Connectionless Mode"; U.S. Patent Application 09/440,693, filed November 16, 1999, entitled "Method and System of Remote Diagnostic, Control and Information Collection Using a Dynamic Linked Library"; U.S. Patent Application 09/440,647, filed November 16, 1999, entitled "Method and System to Monitor the Application Usage and Send Back the Information Using Connection and Connectionless Mode"; U.S. Patent Application 09/440,646, filed November 16, 1999, entitled "Method and System to Monitor the Application Usage and Send Back the Information Using Connection and Connectionless Mode"; U.S. Patent Application 09/440,645, filed November 16, 1999, entitled "Application Unit Monitoring and Reporting System and Method With Usage Data Logged Into a Map Structure"; U.S. Patent Application 09/408,443, filed September 29, 1999, entitled "Method and System for Remote Diagnostic, Control, and Information Collection Based on various Communication Modes for Sending Messages to a Resource Manager"; U.S. Patent Application 09/407,769, filed September 29, 1999, entitled "Method and System for Remote Diagnostic, Control and Information Collection Based on various Communication Modes for Sending Messages to Users"; U.S. Patent Application 09/393,677, filed September 10, 1999, entitled "Application Unit Monitoring and Reporting System and Method"; U.S. Patent Application 09/311,148, filed May 13, 1999, entitled "Application Unit Monitoring and Reporting System and Method"; U.S. Patent Application 09/192,583, filed November 17, 1998, entitled "Method and System for Communicating With a Device Attached to a Computer Using Electronic Mail Messages"; U.S. Patent Application 08/883,492, filed June 26, 1997, entitled "Method and System for Diagnosis and Control of Machines Using Connectionless Modes Having Delivery Monitoring and an Alternate Communication Mode"; U.S. Patent Application 08/820,633, filed March 19, 1997, entitled "Method and System to Diagnose a Business Office Device Based on Operating Parameters Set by a User," now U.S.P. 5,887,216; U.S. Patent Application 08/733,134, filed October 16, 1996, entitled "Method and System for Diagnosis and Control of Machines Using Connectionless Modes of Communication," now U.S.P. 5,909,493; U.S. Patent Application 08/880,683, filed June 23, 1997, U.S. Patent Applications 09/107,989 and 09/108,705, both of which were filed July 1, 1998, all three of which are entitled "Method and System for Controlling and Communicating, with Machines Using Multiple Communication Formats," and all three of which are divisions of U.S. Patent Application 08/624,228, filed March 29, 1996, entitled "Method and System for Controlling and Communicating with Machines Using Multiple Communication Formats," now U.S.P. 5,818,603; U.S. Patent Application 09/457,669, entitled "Method and System for Diagnosis and Control of Machines Using Connection and Connectionless Modes of Communication," filed December 9, 1999, which is a continuation of U.S. Patent Application 08/916,009, entitled "Method and System

for Diagnosis and Control of Machines Using Connection and Connectionless Modes of Communication," filed August 21, 1997, which is a continuation of U.S. Patent Applications 08/738,659 and 08/738,461, filed October 30, 1996, both of which are entitled "Method and System for Diagnosis and Control of Machines Using Connection and Connectionless Modes of Communication," which are divisions of U.S. Patent Application 08/463,002, filed June 5, 1995, entitled "Method and System for Diagnosis and Control of Machines Using Connection and Connectionless Modes of Communication", now U.S.P. 5,819,110; U.S. Patent Application 08/852,413, filed May 7, 1987, entitled "Method and System for Controlling and Communicating with Business Office Devices," now U.S.P. 5,774,678, which is a continuation of U.S. Patent Application 08/698,068, filed August 15, 1996, entitled "Method and Apparatus for Controlling and Communicating With Business Office Devices", now U.S.P. 5,649,120, which is a continuation of U.S. Patent Application 08/562,192, filed November 22, 1995, now U.S.P. 5,568,618, entitled "Method and Apparatus for Controlling and Communicating With Business Office Devices", which is a continuation of U.S. Patent Application 08/473,780, filed June 6, 1995, entitled "Method and Apparatus for Controlling and Communicating With Business Office Devices", now U.S.P. 5,544,289, which is a continuation of U.S. Patent Application 08/426,679, filed April 24, 1995, entitled "Method and Apparatus for Controlling and Communicating With Business Office Devices," now U.S.P. 5,537,554, which is a continuation of U.S. Patent Application 08/282,168, filed July 28, 1994, entitled "Method and Apparatus for Controlling and Communicating With Business Office Devices", now U.S. Patent 5,412,779, which is a continuation of U.S. Patent Application 07/902,462, filed June 19, 1992, now abandoned, which is a continuation of U.S. Patent Application 07/549,278, filed July 6, 1990, now abandoned; U.S. Patent Application XX/XXX,XXX, filed XXX XX, 2001, entitled "SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR SENDING PERSISTENT AND NON-PERSISTENT STATUS INFORMATION TO A MONITOR USING E-MAIL" (Attorney Docket Number 205828US-5244-5244-2); and U.S. Patent Application XX/XXX,XXX, filed XXX XX, 2001, entitled "SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR TRANSFERRING REMOTE DEVICE SUPPORT DATA TO A MONITOR USING E-MAIL" (Attorney Docket Number 205840US-5244-5244-2), the entire contents of each of these applications and patents are incorporated herein by reference.

BACKGROUND OF THE INVENTION

Field of the Invention:

[0002] This invention relates to systems, methods, and computer program products for monitoring networked devices. More particularly, to systems, methods, and computer program products for centrally monitoring devices connected to multiple networks while allowing for network address reuse among networks and network address-based monitoring within a network.

Discussion of the Background:

[0003] Co-pending U.S. patent application Serial No. 09/756,120 filed on January 09, 2001, describes a system for remotely monitoring networked devices using e-mail. As described in that application, the Simple Network Management Protocol (SNMP) is used by a remote monitor to collect information from networked devices. The collected information is then sent to a central monitor using, for example, e-mail.

[0004] Devices connected to a network may be monitored by sending SNMP commands to the various devices based on their IP addresses. The devices will then respond to the SNMP queries with the requested information. The IP addresses of the devices are used to uniquely identify each device on a network. However, an IP address is not necessarily a globally-unique identifier for a particular device. For example, it is known to use network address translators (NAT) as an approach to reusing IP addresses on different networks. A NAT allows a network to have one or more globally-unique addresses by which they are known to the outside world (e.g., the Internet), without requiring that each node of the network behind the NAT have a globally unique IP address. This approach allows, for example, a company to reuse IP addresses on their different internal networks while still providing one or more globally-unique identifier(s) to the outside world.

[0005] These addressing complexities present difficulties to applications that wish to maintain information on the individual devices of a plurality of networks that may or may not share IP addresses. It would be desirable to have the ability to monitor network devices connected to multiple networks from a common central monitoring location. However, device address ambiguities increases the complexity of providing such a capability.

SUMMARY OF THE INVENTION

[0006] The inventors of the present invention have recognized that it would be advantageous for devices connected to multiple networks to have globally-unique identifiers that would enable them to be monitored from a central monitor

that may or may not be connected to one of the networks having devices to be monitored. The inventors of the present invention have recognized that it would be further advantageous if IP address reuse and the ability to monitor devices within a network via SNMP were not sacrificed in providing a solution to device address ambiguity.

[0007] The present invention provides a system, method, and computer program product through which devices residing on multiple networks may be locally monitored using, for example, IP-based SNMP, and centrally monitored without device address ambiguity, and which allows IP address reuse across networks including devices to be monitored by a central location. In one embodiment of the present invention, a remote monitoring workstation monitors devices connected to a network using IP-based SNMP commands and stores the device monitoring information in a database. The remote monitoring workstation queries the database to obtain an IP address and the collected monitoring information for a device, uses the IP address to request a globally-unique identifier (e.g., a MAC address) from a device using SNMP commands, then sends the monitoring information including the globally-unique device identifier to a central monitoring workstation via an e-mail message.

[0008] One advantage of the present invention is that devices from multiple networks may be monitored from a central location without the need for additional processing to resolve device address ambiguities. Another advantage of the present invention is the devices connected to any particular network may continue to be monitored using an IP-based protocol, such as SNMP. Moreover, since address ambiguities are resolved by, for example, a MAC address, prior to sending the monitoring information to a central monitor, there is no requirement that the devices being monitored have globally-unique IP addresses, thereby allowing the present invention to be used in networks that include, for example, NATs.

[0009] Consistent with the title of this section, the above summary is not intended to be an exhaustive discussion of all the features or embodiments of the present invention. A more complete, although not necessarily exhaustive, description of the features and embodiments of the invention is found in the section entitled "DESCRIPTION OF THE PREFERRED EMBODIMENTS."

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] A more complete appreciation of the invention and many of the attendant advantages thereof will be readily obtained as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, wherein:

Figure 1 illustrates three networked business office devices connected to a network of computers and databases through the Internet;

Figure 2 illustrates the components of a digital image forming apparatus;

Figure 3 illustrates the electronic components of the digital image forming apparatus illustrated in Figure 2;

Figure 4 illustrates details of a multi-port communication interface illustrated in Figure 3;

Figure 5 illustrates an alternative system configuration in which business office devices are either connected directly to the network or connected to a computer which is connected to the network;

Figure 6A is a block diagram illustrating a flow of information to and from an application unit using electronic mail;

Figure 6B illustrates an alternative way of communicating using electronic mail in which a computer that is connected to the application unit also serves as a Message Transfer Agent (MTA);

Figure 6C illustrates an alternative way of communicating using electronic mail in which an application unit includes a message transfer agent for exchanging electronic mail;

Figure 6D illustrates an alternative way of communicating using electronic mail in which a mail server acts as a POP3 server to receive mail for an appliance/device and as a Simple Mail Transfer Protocol (SMTP) server to send mail for the appliance/device;

Figure 7 illustrates an alternative manner of sending messages across the Internet;

Figure 8 illustrates an exemplary computer which may be connected to an appliance/device and used to communicate electronic mail messages;

Figure 9 illustrates an overall system configuration related to the present invention;

Figure 10A illustrates a general software architecture of a message sending module;

Figure 10B illustrates a general software architecture of a message receiving module;

Figure 11 illustrates a general architecture of a message sending module;

Figure 12 illustrates a general architecture of a message receiving module;

Figure 13A is a flowchart illustrating a process implemented by the device information module shown in Figure 11;

Figure 13B illustrates a class structure of the device information module;

Figure 14 is a collaboration diagram for the device information module;

Figure 15A is a flowchart illustrating a process implemented by the device monitor module shown in Figure 11;

Figure 15B illustrates a class structure of the device monitor module;

Figures 16, 17 and 18 are collaboration diagrams for the device monitor module;
 Figure 19A is a flowchart illustrating a process implemented by the data transfer module shown in Figure 11;
 Figure 19B is a flowchart illustrating a process for sending information according to one embodiment of the present invention;
 Figure 19C illustrates a class structure of the data transfer module;
 Figures 20A, 21, 22, and 23 are collaboration diagrams for the data transfer module in transferring information to monitoring site;
 Figure 20B illustrates an exemplary MIME attachment including configuration information according to one embodiment of the present invention;
 Figure 20C illustrates an exemplary MIME attachment including status information according to one embodiment of the present invention; and
 Figure 24 illustrates a class diagram of the Open Database Connectivity (ODBC) interface module.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0011] Referring now to the drawings, and more particularly to Figure 1 thereof, there are illustrated (1) various machines and (2) computers for monitoring, diagnosing and controlling the operation of the machines. In Figure 1, there is a first network 16, such as a Local Area Network (LAN) connected to computer workstations 17, 18, 20 and 22. The workstations can be any type of computers including, e.g., IBM Personal Computer compatible devices, Unix-based computers, Linux-based computers or Apple Macintoshes. Also connected to the network 16 are (1) a digital image forming apparatus 24, (2) a facsimile machine 28, and (3) a printer 32. As would be appreciated by one of ordinary skill in the art, two or more of the components of the digital image forming apparatus 24 and the facsimile machine 28 can be combined into a unified "image forming apparatus." The devices 24, 28 and 32 and the workstations 17, 18, 20 and 22 are referred to as machines or monitored devices and other types of devices may be used as the machines or monitored devices, including any of the devices discussed below. In some configurations, one or more workstations may be converted to business office appliances. One example of such a business office appliance is eCabinet from Ricoh, which was demonstrated at Fall Comdex in 1999 at Las Vegas. Also, a facsimile server (not illustrated) may be connected to the network 16 and have a telephone, Integrated Services Digital Network (ISDN), cable or wireless connection. In addition to the digital image forming apparatus 24, facsimile machine 28, and printer 32 being connected to the network 16, these devices may also include conventional telephone and/or ISDN and/or cable and/or wireless connections 26, 30 and 34, respectively. As is explained below, the business office machines, business devices, or business office appliances 24, 28 and 32 communicate with a remote monitoring, diagnosis, and control station, also referred to as a monitoring device, through, for example, the Internet via the network 16 or by a direct telephone, ISDN, wireless, or cable connection.

[0012] In Figure 1, a wide area network (WAN) (e.g., the Internet or its successor) is generally designated by 10. The WAN 10 can be either a private WAN, a public WAN or a hybrid. The WAN 10 includes a plurality of interconnected computers and routers designated by 12A-12I. The manner of communicating over a WAN is known through a series of Request for Comments (RFC) documents available from the Internet Engineering Task Force (IETF) at <http://www.ietf.org/rfc.html>, including RFC 821 entitled "Simple Mail Transfer Protocol"; RFC 822 entitled "Standard for the Format of ARPA Internet Text Message"; RFC 959 entitled "File Transfer Protocol (FTP)"; RFC 2045 entitled "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies"; RFC 1894 entitled "An Extensible Message Format for Delivery Status Notifications"; RFC 1939 entitled "Post Office protocol - Version 3"; and RFC 2298 entitled "An Extensible Message Format for Message Disposition Notifications." The contents of each of these references are incorporated herein by reference.

[0013] Transmission Control Protocol/Internet Protocol (TCP/IP) related communication is described, for example, in the book "TCP/IP Illustrated," Vol. 1, The Protocols, by W.R. Stevens, from Addison-Wesley Publishing Company, 1994, the entire contents of which is incorporated herein by reference. Volumes 1-3 of "Internetworking with TCP/IP" by Comer and Stevens are also incorporated herein by reference in their entirety.

[0014] In Figure 1, a firewall 50A is connected between the WAN 10 and the network 16. A firewall is a device that allows only authorized computers on one side of the firewall to access a network, computers or individual parts on the other side of the firewall. Firewalls are known and commercially available devices and/or software (e.g., SunScreen from Sun Microsystems Inc.). Similarly, firewalls 50B and 50C separate the WAN 10 from a network 52 and a workstation 42, respectively. Additional details on firewalls can be found in "Firewalls and Internet Security" by W. R. Cheswick, and S. M. Bellovin, 1994, AddisonWesley Publishing, and "Building Internet Firewalls" by D. B. Chapman and E. D. Zwicky, 1995, O'Reilly & Associates, Inc. The entire contents of those two references are incorporated herein by reference.

[0015] The network 52 is a conventional network and includes a plurality of workstations 56, 62, 68 and 74. These workstations may be in different departments (e.g., marketing, manufacturing, design engineering, and customer serv-

ice departments) within a single company. In addition to the workstations connected via the network 52, there is a workstation 42, which is not directly connected to the network 52. Information in a database stored in a disk 46 may be shared using proper encryption and protocols over the WAN 10 to the workstations connected directly to the network 52. Also, the workstation 42 includes a direct connection to a telephone line and/or an ISDN and/or a cable and/or a wireless network 44, and the database in disk 46 may be accessed through the telephone line, the ISDN, the cable or wirelessly. The cable used by this invention may be implemented using a cable that is typically used to carry television programming, a cable that provides for high-speed communication of digital data typically used with computers or the like, or any other desired type of cable.

[0016] Information of the business office machines, business devices or business office appliances 24, 28 and 32 may be stored in one or more of the databases stored in the disks 46, 54, 58, 64, 70 and 76. Known databases include (1) SQL databases by Microsoft, IBM, Oracle and Sybase (2) other relational databases, and (3) non-relational databases (including object oriented databases from Computer Associates, JYD Software Engineering, and Orient Technologies). Each of the customer service, marketing, manufacturing, and engineering departments may have their own database or may share one or more databases. Each of the disks used to store databases is a non-volatile memory such as a hard disk or optical disk. Alternatively, the databases may be stored in any storage device including solid state and/or semiconductor memory devices. As an example, disk 64 contains the marketing database, disk 58 contains the manufacturing database, disk 70 contains the engineering database and disk 76 contains the customer service database. Alternatively, the disks 54 and 46 store one or more of the databases.

[0017] In addition to the workstations 56, 62, 68, 74 and 42 being connected to the WAN 10, these workstations may also include a connection to a telephone line, ISDN, cable, or wireless network which provides a secure connection to the machine being monitored, diagnosed and/or controlled, and is used during communication. Additionally, if one communication medium is not operating properly, one of the others can be automatically used for communication.

[0018] A feature of the present invention is the use of a "store-and-forward" mode of communication (e.g., Internet electronic mail, also referred to herein as e-mail) or transmission between a machine and a computer for diagnosing and controlling the machine. Alternatively, the message which is transmitted may be implemented using a mode of communication that makes direct, end-to-end connections (e.g., using a socket connection to the ultimate destination) such as FTP and Hyper Text Transfer Protocol (HTTP).

[0019] Figure 2 illustrates the mechanical layout of the digital image forming apparatus 24 illustrated in Figure 1. In Figure 2, 101 is a fan for the scanner, 102 is a polygonal mirror used with a laser printer, and 103 designates an Fθ lens used to collimate light from a laser (not illustrated). Reference numeral 104 designates a sensor for detecting light from the scanner. Reference numeral 105 designates a lens for focusing light from the scanner onto the sensor 104, and reference numeral 106 designates a quenching lamp used to erase images on the photoconductive drum 132. There is a charging corona unit 107 and a developing roller 108. Reference numeral 109 designates a lamp used to illustrate a document to be scanned and 110, 111, and 112 designate mirrors used to reflect light onto the sensor 104. There is a drum mirror 113 used to reflect light to the photoconductive drum 132 originating from the polygon mirror 102. Reference numeral 114 designates a fan used to cool the charging area of the digital image forming apparatus, and reference numeral 115 designates a first paper feed roller used for feeding paper from the first paper cassette 117, and reference numeral 116 designates a manual feed table. Similarly, reference numeral 118 designates a second paper feed roller for the second cassette 119. Reference numeral 120 designates a relay roller, 121 designates a registration roller, 122 designates an image density sensor, and 123 designates a transfer/separation corona unit. Reference numeral 124 designates a cleaning unit, 125 designates a vacuum fan, 126 designates a transport belt, 127 designates a pressure roller, and 128 designates an exit roller. Reference numeral 129 designates a hot roller used to fix toner onto the paper, 130 designates an exhaust fan and 131 designates the main motor used to drive the digital image forming apparatus.

[0020] Figure 3 is a block diagram illustrating the electronic components of the digital image forming apparatus of Figure 2. The CPU 160 is a microprocessor and acts as the system's controller. Random access memory (RAM) 162 stores dynamically changing information including operating parameters of the digital image forming apparatus. A non-volatile memory (e.g., a read only memory 10 (ROM) 164 or a Flash Memory) stores (1) the program code used to run the digital image forming apparatus and (2) static-state data, describing the copier (e.g., the model number, serial number of the copier, and default parameters).

[0021] There is a multi-port network interface 166 which allows the digital image forming apparatus to communicate with external devices through at least one network. Reference number 168 represents a telephone, ISDN, or cable line, and numeral 170 represents another type of network. Additional details of the multi-port network interface are described with respect to Figure 4. An interface controller 172 is used to connect an operation panel 174 to a system bus 186. The operation panel 174 includes standard input and output devices found on a digital image forming apparatus including a copy button, keys to control the operation of the copier such as number of copies, reduction/enlargement, darkness/lightness, etc. Additionally, a liquid crystal display may be included within the operation panel 174 to display parameters and messages of the digital image forming apparatus to a user.

[0022] A local connection interface 171 is a connection through local ports such as RS232, the parallel printer port, USB, and IEEE 1394. FireWire (IEEE 1394) is described in Wickelgren, I., "The Facts About "FireWire", IEEE Spectrum, April 1997, Vol. 34, Number 4, pp. 19-25, the entire contents of which are incorporated herein by reference. Preferably, a "reliable" communication protocol is used which includes error detection and retransmission.

5 [0023] A storage interface 176 connects storage devices to the system bus 186. The storage devices include a flash memory 178, which can be substituted by a conventional Electrically Erasable Programmable Read Only Memory (EEPROM), and a disk 182. The disk 182 includes a hard disk, optical disk, and/or a floppy disk drive. There is a connection 180 connected to the storage interface 176 which allows for additional memory devices to be connected to the digital image forming apparatus. The flash memory 178 is used to store semi-static state data which describes
10 parameters of the digital image forming apparatus which infrequently change over the life of the copier. Such parameters include the options and configuration of the digital image forming apparatus. An option interface 184 allows additional hardware such as an external interface to be connected to the digital image forming apparatus. A clock/timer 187 is utilized to keep track of both the time and date and also to measure elapsed time.

[0024] On the left side of Figure 3, the various sections making up the digital image forming device are illustrated.
15 Reference numeral 202 designates a sorter and contains sensors and

actuators used to sort the output of the digital image forming device. There is a duplexer 200 which allows a duplex operation to be performed by the digital image forming device and includes conventional sensors and actuators. The digital image forming device includes a large capacity tray unit 198 which allows paper trays holding a large number of sheets to be used with the digital image forming device. The large capacity tray unit 198 includes conventional
20 sensors and actuators.

[0025] A paper feed controller 196 is used to control the operation of feeding paper into and through the digital image forming device. A scanner 194 is used to scan images into the digital image forming device and includes conventional scanning elements such as a light, mirror, etc. Additionally, scanner sensors are used such as a home position sensor to determine that the scanner is in the home position, and a lamp thermistor is used to ensure proper operation of the
25 scanning lamp. There is a printer/imager 192 which prints the output of the digital image forming device, and includes a conventional laser printing mechanism, a toner sensor, and an image density sensor. The fuser 190 is used to fuse the toner onto the page using a high temperature roller and includes an exit sensor, a thermistor to assure that the fuser 190 is not overheating, and an oil sensor. Additionally, there is an optional unit interface 188 used to connect to optional elements of the digital image forming device such as an automatic document feeder, a different type of sorter/
30 collator, or other elements which can be added to the digital image forming device.

[0026] Figure 4 illustrates details of the multi-port network interface 166. The digital image forming device may communicate to external devices through a token ring interface 220, a cable modem unit 222, which has a high speed connection over cable, a conventional telephone interface 224, which connects to a telephone line 168A, an ISDN interface 226, which connects to an ISDN line 168B, a wireless interface 228, or an ethernet interface 230, which
35 connects to a LAN 170. Other interfaces may include, but are not limited to, a Digital Subscriber Line (DSL) (original DSL, concentric DSL, and asymmetric DSL). A single device which connects to both a Local Area Network and a telephone line is commercially available from Megahertz and is known as the Ethernet-Modem.

[0027] The CPU or other microprocessor or circuitry executes a monitoring process to monitor the state of each of the sensors of the digital image forming device, and a sequencing process is used to execute the instructions of the code used to control and operate the digital image forming device. Additionally, there is (1) a central system control process executed to control the overall operation of the digital image forming device, and (2) a communication process used to assure reliable communication to external devices connected to the digital image forming device. The system control process monitors and controls data storage in a static state memory (e.g., the ROM 164 of Figure 3), a semi-static memory (e.g., the flash memory 178 or disk 182), or the dynamic state memory (e.g., a volatile or non-volatile
45 memory (e.g., the RAM 162 or the flash memory 178 or disk 182). Additionally, the static state memory may be a device other than the ROM 164 such as a non-volatile memory including either of the flash memory 178 or disk 182.

[0028] The above details have been described with respect to a digital image forming device, but the present invention is equally applicable to other business office machines or devices such as an analog copier, a facsimile machine, a scanner, a printer, a facsimile server, or other business office machines, a business office appliance, or other appliances
50 (e.g., a microwave oven, VCR, digital camera, cellular phone, palm top computer). Additionally, the present invention includes other types of devices which operate using store-and-forward or direct connection-based communication. Such devices include metering systems (including gas, water, or electricity metering systems), vending machines, or any mechanical device (e.g., automobiles) that needs to be monitored during operation or remote diagnosis. In addition to monitoring special purpose machines and computers, the invention can be used to monitor, control, and diagnose
55 a general purpose computer that would be the monitored and/or controlled device.

[0029] Figure 5 illustrates an alternative system diagram of the present invention in which different devices and subsystems are connected to the WAN 10. However, there is no requirement to have each of these devices or subsystems as part of the invention. Each component or subsystem illustrated in Figure 5 is individually part of the invention.

Further, the elements illustrated in Figure 1 may be connected to the WAN 10 which is illustrated in Figure 5. In Figure 5, there is illustrated a firewall 50-1 connected to an intranet 260-1. A service machine 254 connected to the intranet 260-1 includes therein, or has connected thereto, data 256 that may be stored in a database format. The data 256 includes history, performance, malfunction, and any other information such as statistical information of the operation or failure or set-up the monitored devices, or configuration information such as which components or optional equipment is included with the monitored devices. The service machine 254 may be implemented as the device or computer which requests the monitored devices to transmit data, or which requests that remote control and/or diagnostic tests be performed on the monitored devices. The service machine 254 may be implemented as any type of device, and is preferably implemented using a computerized device such as a general purpose computer.

[0030] Another sub-system of Figure 5 includes a firewall 50-2, an intranet 260-2, and a printer 262 connected thereto. In this sub-system, the functions of sending and receiving electronic messages by the printer 262 (and similarly by a copier 286) are performed by (1) circuitry, (2) a microprocessor, or (3) any other type of hardware contained within or mounted to the printer 262 (i.e., without using a separate general purpose computer).

[0031] An alternate type of sub-system includes the use of an Internet Service Provider 264 which maybe any type of Internet Service Provider (ISP), including known commercial companies such as America Online, Earthlink, and Niftyserve. In this sub-system, a computer 266 is connected to the ISP 264 through a digital or analog modem (e.g., a telephone line modem, a cable modem, modems which use any type of wires such as modems used over an Integrated Services Digital Network (ISDN) line or an Asymmetric Digital Subscriber Line (ADSL), modems which use frame relay communication, wireless modems such as a radio frequency modem, a fiber optic modem, or a device which uses infrared light waves). Further, a business office device 268 is connected to the computer 266. As an alternative to the business office device 268 (or any other device illustrated in Figure 5), a different type of machine may be monitored or controlled such as a digital copier, any type of appliance, security system, or utility meter, such as an electrical, water, or gas utility meter, or any other device discussed herein.

[0032] Also illustrated in Figure 5 is a firewall 50-3 connected to a network 274. The network 274 may be implemented as any type of computer network, (e.g., an ethernet or token ring network). Networking software which may be used to control the network includes any desired networking software including software commercially available from Novell or Microsoft. The network 274 may be implemented as an intranet, if desired. A computer 272 connected to the network 274 may be used to obtain information from a business office device 278 and generate reports such as reports showing problems that occurred in various machines connected to the network, and a monthly usage report of the devices connected to the network 274. In this embodiment, a computer 276 is connected between the business office device 278 and the network 274. This computer receives communications from the network and forwards the appropriate commands or data, or any other information, to the business office device 278. Communication between the business office device 278 and the computer 276 may be accomplished using wire-based or wireless methods including, but not limited to, radio frequency connections, electrical connections and light connections (e.g., an infrared connection, or a fiber optics connection). Similarly, each of the various networks and intranets illustrated in Figure 5 may be established using any desired manner including through the establishment of wireless networks such as radio frequency networks. The wireless communication described herein may be established using spread spectrum techniques including techniques which use a spreading code and frequency hopping techniques such as the frequency hopping wireless technique which is disclosed in the Bluetooth Specification LOA (available at the World Wide Web site www.bluetooth.com), which is incorporated herein by reference.

[0033] Another sub-system illustrated in Figure 5 includes a firewall 50-4, an intranet 260-4, a computer 282 connected thereto, a business office appliance 285 and a copier 286. The computer 282 may be used to generate reports and request diagnostic or control procedures. These diagnostic and control procedures may be performed with respect to the business office appliance 285 and the copier 286 or any of the other devices illustrated in or used with Figure 5. While Figure 5 illustrates a plurality of firewalls, the firewalls are preferable, but optional equipment, and therefore, the invention may be operated without the use of firewalls, if desired.

[0034] Figure 6A illustrates a device/appliance 300 connected to a typical e-mail exchange system which includes components 302, 304, 306, 308, 310, 312, 314, 316, and 318, which may be implemented in a conventional manner, and are adapted from Figure 28.1 of Stevens, above. A computer interface 302 interfaces with any of the application units or devices/appliances 300 described herein. While Figure 6A illustrates that the device/appliance 300 is the sender, the sending and receiving functions may be reversed in Figure 6A. Furthermore, if desired, the user may not be needed to interface with the device/appliance 300 at all. The computer interface 302 then interacts with a mail agent 304. Popular mail agents for Unix include MH, Berkeley Mail, Elm, and Mush. Mail agents for the Windows family of operating systems include Microsoft Outlook and Microsoft Outlook Express. At the request of the computer interface 302, the mail agent 304 creates e-mail messages to be sent and, if desired, places these messages to be sent in a queue 306. The mail to be sent is forwarded to a Message Transfer Agent (MTA) 308. A common MTA for Unix systems is Sendmail. Typically, the message transfer agents 308 and 312 exchange communications using a TCP/IP connection 310. Notably, the communication between the message transfer agents 308 and 312 may occur over any size network

(e.g., WAN or LAN). Further, the message transfer agents 308 and 312 may use any communication protocol. In one embodiment of the present invention, elements 302 and 304 of Figure 6A reside in the library to monitor the usage of the application unit.

[0035] From the message transfer agent 312, e-mail messages are stored in user mailboxes 314 which are transferred to, the mail agent 316 and ultimately transmitted to the user at a terminal 318 which functions as a receiving terminal.

[0036] This "store-and-forward" process relieves the sending mail agent 304 from having to wait until a direct connection is established with the mail recipient. Because of network delays, the communication could require a substantial amount of time during which the application would be unresponsive. Such an unresponsiveness is generally unacceptable to users of the application unit. By using e-mail as the store-and-forward process, retransmission attempts after failures occur automatically for a fixed period of time (e.g., three days). In an alternate embodiment, the application can avoid waiting by passing communicating requests to one or more separate threads. Those threads can then control communication with the receiving terminal 318 while the application begins responding to the user interface again. In yet another embodiment in which a user wishes to have communication completed before continuing, direct communication with the receiving terminal is used. Such direct communication can utilize any protocol not blocked by a firewall between the sending and receiving terminals. Examples of such protocols include File Transfer Protocol (FTP) and Hyper Text Transfer Protocol (HTTP).

[0037] Public WANs, such as the Internet, are generally not considered to be secure. Therefore, if it is desired to keep messages confidential, messages transmitted over the public WANs (and multi-company private WANs) can be encrypted. Encryption mechanisms are known and commercially available which may be used with the present invention. For example, a C++ library function, crypt(), is available from Sun Microsystems for use with the Unix operating system. Other encryption and decryption software packages are known and commercially available and may also be used with this invention. One such package is PGP Virtual Private Network (VPN) available from Network Associates. Other VPN software is available from Microsoft Corporation.

[0038] As an alternative to the general structure of Figure 6A, a single computer may be used which functions as the computer interface 302, the mail agent 304, the mail queue 306 and the message transfer agent 308. As illustrated in Figure 6B, the device/appliance 300 is connected to a computer 301 which includes the message transfer agent 308.

[0039] A further alternative structure is shown in Figure 6C in which the message transfer agent 308 is formed as part of the device/appliance 300. Further, the message transfer agent 308 is connected to the message transfer agent 312 by a TCP/IP connection 310. In the embodiment of Figure 6C, the device/appliance 300 is directly connected to the TCP/IP connection 310 with an e-mail capability. One use of the embodiment of Figure 6C includes using a facsimile machine with an e-mail capability (e.g., as defined in RFC 2305 (a simple mode of facsimile using Internet mail)) as the device/appliance 300.

[0040] Figure 6D illustrates a system in which a device/appliance 300 does not by itself have the capability to directly receive e-mail, but has a connection 310 to a mail server/POP3 server including a message transfer agent 308 and a mail box 314 so that the device/appliance 300 uses the POP3 protocol to retrieve received mail from the mail server.

[0041] Figure 7 illustrates an alternative implementation of transferring mail and is adapted from Figure 28.3 of Stevens referenced previously. Figure 7 illustrates an electronic mail system having a relay system at each end. The arrangement of Figure 7 allows one system at an organization to act as a mail hub. In Figure 7, there are four MTAs connected between the two mail agents 304 and 316. These MTAs include local MTA 322A, relay MTA 328A, relay MTA 328B, and local MTA 322D. The most common protocol used for mail messages is SMTP (Simple Mail Transfer Protocol) which may be used with this invention, although any desired mail protocol may be utilized. In Figure 7, 320 designates a sending host which includes the computer interface 302, the mail agent 304, and the local MTA 322A. The device/appliance 300 is connected to, or alternatively included within, the sending host 320. As another case, the device/appliance 300 and host 320 can be in one machine where the host capability is built into the device/appliance 300. Other local MTAs 322B, 322C, 322E and 322F may also be included. Mail to be transmitted and received may be queued in a queue of mail 306B of the relay MTA 328A. The messages are transferred across the TCP/IP connection 310 (e.g., an Internet connection or a connection across any other type of network).

[0042] The transmitted messages are received by the relay MTA 328B and if desired, stored in a queue of mail 306C. The mail is then forwarded to the local MTA 322D of a receiving host 342. The mail may be placed in one or more of the user mailboxes 314 and subsequently forwarded to the mail agent 316, and finally forwarded to the user at a terminal 318. If desired, the mail may be directly forwarded to the terminal without user interaction.

[0043] The various computers used in the present invention, including the computers 266 and 276 of Figure 5, may be implemented as illustrated in Figure 8. Further, any other computer used in this invention may be implemented in a similar manner to the computer illustrated in Figure 8, if desired, including the service machine 254, computer 272, and computer 282 of Figure 5. However, not every element illustrated in Figure 8 is required in each of those computers.

[0044] In Figure 8, the computer 360 includes a CPU 362 which may be implemented as any type of processor including commercially available microprocessors from companies such as Intel, AMD, Motorola, Hitachi and NEC.

There is a working memory such as a RAM 364, and a wireless interface 366 which communicates with a wireless device 368. The communication between the interface 366 and device 368 may use any wireless medium (e.g., radio waves or light waves). The radio waves may be implemented using a spread spectrum technique such as Code Division Multiple Access (CDMA) communication or using a frequency hopping technique such as that disclosed in the Bluetooth specification.

[0045] There is a ROM 370 and a flash memory 371, although any other type of non-volatile memory (e.g., Erasable Programmable ROM, or an EEPROM) may be used in addition to or in place of the flash memory 371. An input controller 372 has connected thereto a keyboard 374 and a mouse 376. There is a serial interface 378 connected to a serial device 380. Additionally, a parallel interface 382 is connected to a parallel device 384, a universal serial bus (USB) interface 386 is connected to a universal serial bus device 388, and also there is an IEEE 1394 device 400, commonly referred to as a fire wire device, connected to an IEEE 1394 interface 398. The various elements of the computer 360 are connected by a system bus 390. A disk controller 396 is connected to a floppy disk drive 394 and a hard disk drive 392. A communication controller 400 allows the computer 360 to communicate with other computers (e.g., by sending e-mail messages) over a telephone line 402 or a network 404. An I/O (Input/Output) controller 408 is connected to a printer 410 and a hard disk 412, for example using a SCSI (Small Computer System Interface) bus. There is also a display controller 416 connected to a CRT (Cathode Ray Tube) 414, although any other type of display may be used including a liquid crystal display, a light emitting diode display, a plasma display, etc.

[0046] Figure 9 illustrates an application of the present invention. Devices 901, 903, 905 and 907 that are connected to the Intranet 910 are the devices to be monitored locally by a remote monitoring workstation 911 with its database 913. Alternatively, the remote monitoring workstation 911 can function to send the device status information to the central monitoring workstation 945 by polling the information from the monitored devices 901, 903, 905, and 907 and by sending the information through the firewall 917. The remote monitoring workstation 911, therefore, can function either as a monitoring device or as a communication and administrative device between the monitored devices and monitoring device. In Figure 9, the remote monitoring workstation 911 uses the Simple Network Management Protocol (SNMP) defined by IETF to communicate with the attached devices. SNMP is described in "Managing Internetworks with SNMP, third edition" by Mark A. Miller, P.E., M & T Book, 1999. The entire contents of this reference are incorporated herein by reference. If some of the devices to be monitored do not support SNMP, the remote monitoring workstation 911 can use a different method to obtain the necessary information. After obtaining the necessary information, the remote monitoring workstation 911 uses the Simple Mail Transfer Protocol (SMTP) Server 915 to send out the necessary information to the central monitoring workstation 945 through the Mail Server 943 that supports the Post Office Protocol Version 3 (POP3) (IETF Networking Group Request For Comments [RFC]: 1939). The remote monitoring workstation 911 uses SMTP (SMTP is defined in IETF RFC 821) and possibly Multipurpose Internet Mail Extensions (MIME) to send e-mails. The remote monitoring workstation 911 generates the mail message that is at and above the Application Layer of the TCP/IP model or the ISO seven-layer model, as shown later. Alternatively, the remote monitoring workstation 911 may include an SMTP processor to send out the necessary information using e-mail.

[0047] The LAN 920 and intranet 930 send similar information to the central monitoring workstation 945. When the e-mails that contain the monitoring information of devices arrive at the firewall 941 of the intranet 950, the mail is routed to the Mail Server 943 with POP3. The central monitoring workstation 945 periodically accesses the Mail Server 943 to obtain the arrived e-mail, parse the mail and its content via POP3 and stores the necessary information in the database 947. The database 949 contains the additional information of the monitored device characteristics and history. The computers 951 and 953 perform the analysis of obtained data to take the necessary actions. Alternatively, the central monitoring workstation 945 may contain a mail receiving capability, and the firewall may route the e-mail directly to the central monitoring workstation 945.

[0048] Figures 10A and 10B illustrate an overall software architecture of the system shown in Figure 9 according to one embodiment of the present invention. Figure 10A illustrates the architecture of the software used by the networks that send the e-mails with the information on the monitored devices in Figure 9 according to one embodiment of the present invention. The Sender Service 1001 module is the system resident software that sets up the destination for the monitored information to be sent, initiates the sending of the configuration and contact information to the destination, and periodically monitors and sends the information to the destination by using the three functions defined in 1000 (i.e., setDestination, obtainAndUpdateStatus, and sendConfig) to trigger the send module, Monitor_Send DLL 1003. The Monitor_Send DLL 1003 module uses two other modules, the Database 1005 module to store the device information and device-related information along with the monitored information that is stored until it is sent out, and the SNMP++ DLL 1007 module that is used to obtain the information from the devices.

[0049] Figure 10B illustrates the architecture of the software used by the receiving side (e.g., intranet 950) in Figure 9 according to one embodiment of the present invention. The Receiver Service 1011 module is the system resident software that sets up access to the mail server where the monitored information is to be sent, and periodically obtains the monitored information from the mail server through the two functions defined in 1010 (i.e., setupPOP3Server, and getMailAndUpdateDatabase) to trigger the Receive_Store DLL 1013 module. The Receive_Store DLL 1013 module

uses two other modules, the Database 1017 module to store device information and device-related information along with the monitored information and the POP3 1015 module to retrieve information from the mail server.

[0050] Figure 11 illustrates the general architecture of the Monitor_Send DLL 1003 module according to one embodiment of the present invention. This part of the system is responsible for monitoring the status of the devices and for sending e-mails containing status and configuration information of the monitored devices. The Interface 1101 module allows any application to use the Monitor_Send DLL 1003 module. For example, the Sender Service 1001 module in Figure 10A accesses the Monitor_Send DLL 1003 module through the Interface 1101 module. The Device Information 1105 module is responsible for obtaining configuration information from the monitored devices and initiating the sending of the configuration information. The Device Monitor 1103 module is responsible for obtaining status information from the monitored devices and initiating the sending of the status information. The Data Transfer 1107 module is responsible for providing a method through which the status and configuration information is sent. The ODBC Interface 1109 module provides a method to access and store information in a database. Each of the components of the Monitor_Send DLL 1003 module provides interface functions that allow them to perform its tasks. For example, the functions of the Data Transfer 1107 module are provided through four interface functions, setDestination, startSend, dataSend, and end-Send.

[0051] Figure 12 illustrates a general architecture of the Receive_Store DLL 1013 module according to one embodiment of the present invention. This part of the system is responsible for retrieving the information that was sent to it by the Monitor_Send DLL 1003 module and storing the information in the database. The Interface 1101 module allows any application to use the Receive_Store DLL 1013 module. For example, the Receiver Service 1011 module in Figure 10B accesses the Receive_Store DLL 1013 module through the Interface 1101 module. The Receive Manager 1203 module is responsible for obtaining the configuration information and status information of the monitored devices from the POP3 server and storing that information in the database. The Data Retriever 1205 module is responsible for retrieving the data from the POP3 server. The POP3 Processor 1207 module is responsible for accessing the information sent to it by the Monitor_Send DLL 1003 module. The Parser 1211 module is responsible for parsing the information obtained from the POP3 server. The ODBC Interface 1109 module is responsible for storing the information sent to it in a database. Each of the components of the Receive_Store DLL 1013 module provides interface functions that allow them to perform its tasks.

[0052] Figure 13A is a flowchart providing an overview of the functions performed by the Device Information 1105 module in the context of the system diagram of Figure 9. This process focuses on the sending of configuration information of the monitored devices from the remote monitoring workstation 911 to the central monitoring workstation 945, and not the sending of status information, which is described below in the context of the Device Monitor 1103 module. Configuration information for the monitored devices maintained by the central monitor may be either originally sent, or updated, through the functions performed by the Device Information 1105 module, as will be understood in light of the description provided herein.

[0053] As shown in Figure 13A, the process begins with step S 1301 where the database 913 is queried by the remote monitoring workstation 911 to obtain configuration information and IP address information corresponding to the devices being monitored by that particular remote monitoring workstation 911. The process then proceeds to step S1302 where, using the IP address obtained from the database 913, the remote monitoring workstation 911 queries the individual monitored devices using SNMP commands to obtain a device-unique identifier (e.g., a MAC address) for each of the devices being monitored. The process then proceeds to step S1303 where the remote monitoring workstation 911 stores the device-unique identifier in the database. The process then proceeds to step S1304 where the configuration information, including the IP address obtained from the database 913 and the device-unique identifier obtained through SNMP commands, are formatted into a common map structure. The process then proceeds to step S1305 where the configuration information including the device-unique identifier is sent to the central monitoring workstation 945 via an e-mail message through the SMTP server 915. Once the information has been sent, the process ends.

[0054] Figure 13B is a class diagram illustrating one embodiment of the Device Information 1105 module of Figure 11 according to the present invention. The Device Information 1105 module is responsible for triggering the Data Transfer 1107 module to start the sending of configuration information, obtaining the device configuration information from the database 913 via the ODBC Interface 1109 module, obtaining a device-unique identifier (e.g., a MAC address) from the monitored SNMP devices and updating the device configuration information in the database 913 to include the device-unique identifier, formatting the device configuration information into a map structure, sending the map structure to the Data Transfer 1107 module, and completing the sending of the configuration information via the Data Transfer 1107 module.

[0055] The database 913 is not initially populated with a device-unique identifier for the monitored devices. The Device Information 1105 module is responsible for obtaining this information from the monitored devices based on information that is originally stored in the database (e.g., IP address information), via SNMP commands. The device-unique identifier, obtained directly from the devices, is then populated by the Device Information 1105 module into the database 913.

[0056] The map structure used in storing the information to be sent from the remote monitoring workstation 911 to the central monitoring workstation 945 is a standard structure for storing key/value data. Each entry in the map includes a key indicating what the data represents, and a data field containing the value of the data. In one embodiment of the present invention, the key of the map is a string or a number associated with a particular data field, and the data field is a string value. Map structures are included with the standard C++ language, and similar structures, sometimes called a dictionary structure, are included with other standard languages. An example of a populated map structure is shown below as Table 1:

Table 1:

Example Map Structure Including Configuration Information	
Key	Value (Data)
"Manufacturer" (or 100)	"Xerox"
"Model" (or 101)	"DocuPrint N4025"
"SerialNumber" (or 102)	"PF4-027955"
"MACAddress" (or 103)	"00 00 AA 79 07 76"
"IPAddress" (or 104)	"172.30.4.53"
"CompanyName" (or 105)	"Ricoh Corporation"
"Street" (or 106)	"1996 Lundy Ave"
"City" (or 107)	"San Jose"
"State" (or 108)	"CA"
"ZipCode" (or 109)	"95131"
"Location" (or 110)	"Lab"
"ContactPerson" (or 111)	"John Smith"
"PhoneNumber" (or 112)	"4085551212"

[0057] The Device Information 1105 module contains two classes, CDeviceInformation 1301 and CIP_MACmap 1303. The CDeviceInformation 1301 class is responsible for obtaining the configuration information from the database 913, and initiating the sending of the information through e-mail from the remote monitoring workstation 911 to the central monitoring workstation 945. The CDeviceInformation 1301 class interacts with the database 913 via the ODBC Interface 1109 module to obtain the configuration information, and uses the Data Transfer 1107 module to transmit the configuration information to the central monitoring workstation 945.

[0058] The CIP_MACmap 1303 class makes use of the CSnmpResource 1305 class to obtain a physical address (e.g., a MAC address) from the monitored SNMP devices. The MAC address is used to uniquely identify the monitored devices within the database 947 maintained by, for example, the central monitoring workstation 945. While an IP address, for example, may uniquely identify a monitored device among the devices connected to a particular network monitored by the remote monitoring workstation 911, that address may not be unique among all of the networks being monitored by the central monitoring workstation 945. It is for this reason that, in this example, a MAC address is used to provide a globally unique identification for a particular device that can be relied on by the central monitoring workstation 945.

[0059] If other device-unique identification is available, the class structure shown in Figure 13B can be modified to accommodate that unique identification.

[0060] Figure 14 is a collaboration diagram illustrating the interaction among the classes of the Device Information 1105 module shown in Figure 13B to obtain and send configuration information of a monitored device from the remote monitoring workstation 911 to the central monitoring workstation 945. As shown in Figure 14, the process is initiated by a call to the sendConfig() method of the CDeviceInformation 1403 class by the Interface 1101. In response, the CDeviceInformation 1403 class calls the startSend() method of the CDataTransfer 1405 class to initiate a communication link for sending of the e-mail message that will contain the configuration information. The CDeviceInformation 1403 class then calls the getDeviceInformation() method of the CSendODBCInterface 1411 class to obtain the configuration information, including the IP address, of the monitored device from the database. The CDeviceInformation 1403 class then calls the getMACforIP() method of the CIP_MACmap 1407 class to obtain a physical address (e.g., the MAC address) for the monitored devices based on the IP address that was obtained from the database. In turn, the

CIP_MACmap 1407 class calls the setIPAddressOfAgent() and getOctetStringValueForOID() methods of the CSnmpResource 1409 class to query the monitored device based on its IP address to receive its physical address through the appropriate SNMP functions. Next, the CDeviceInformation 1403 class calls the setDeviceInformation() method of the CSendODBCInterface 1411 class to store the configuration information in the database.

5 [0061] The CDeviceInformation 1403 class then calls the dataSend() method of the CDataTransfer 1405 class to send the configuration information, along with the physical address information, to the central monitoring workstation 945. Finally, the CDeviceInformation 1403 class calls the endSend() method of the CDataTransfer 1405 class to complete the sending of the configuration information.

10 [0062] Figure 15A is a flowchart providing an overview of the functions performed by the Device Monitor 1103 module in the context of the system diagram of Figure 9. This process focuses on the collection, storing, and sending of information of the monitored devices from the remote monitoring workstation 911 to the central monitoring workstation 945 as an e-mail message via the SMTP server 915.

15 [0063] As shown in Figure 15A, the process begins with step S1501 where it is determined whether information is to be sent to the central monitoring workstation 945. In one embodiment of the present invention, some information is sent from the remote monitoring workstation 911 to the central monitoring workstation 945 at a different frequency (e.g., less frequently) than a frequency that the monitored devices are polled for status. If it is determined that the collected information is not to be sent to the central monitoring workstation 945 (i.e., "NO" at step S1501), the process proceeds to step S1502 where the monitored devices are polled for a first type of information only.

20 [0064] The first type of information may include, for example, certain status information that may change states more frequently than information is reported to the central monitoring workstation. A second type of information may include a different class of status information, for example, a counter, a level indicator, or a configuration setting of a monitored device. For this second type of information, interim values between reporting periods are not of interest. As would be understood, it is quite possible that, depending on the frequency with which information is sent to the central monitoring workstation 945, status information corresponding to the first type of information, for example, an error condition, could have been corrected between transmissions to the central monitoring workstation 945. For that reason, it is helpful to store the first type of information, so that when information is sent to the central monitoring workstation 945, it can be reported that, in this example, a particular error condition, while not necessarily still present, had occurred since the last time information was sent. Accordingly, when the information, including both the first type and the second type of information, is sent to the central monitoring workstation 945, the first type of information stored in the database 913 is queried from the database 913 and sent along with the most recent information. Then, those values in the database 913 are reset to clear any information that had been stored leading up to the transmission to the central monitoring workstation 945.

25 [0065] Returning to Figure 15A, once the first type of information has been collected from the network device, the process proceeds to step S1503 where the first type of information is stored in the database 913 by the remote monitoring workstation 911. After the first type of information is stored in the database 913, the process ends.

30 [0066] If, on the other hand, it is determined that information is to be sent to the central monitoring workstation 945 (i.e., "YES" at step S1501), the process proceeds to step S1504 where the monitored devices are polled for both the first type of information and the second type of information. Once this information is obtained, the process proceeds to step S1505 where the database is queried for the stored first type of information previously collected. The process then proceeds to step S1506 where both the first type and the second type of information just collected, as well as the first type of information retrieved from the database, is formatted into a common map structure. This is the same map structure that was used by the Device Information 1105 module to send the configuration information. The process then proceeds to step S1507 where both the first type and second type of information is sent by the remote monitoring workstation 911 to the central monitoring workstation 945 as an e-mail message via the SMTP server 915. After the status information has been sent by the remote monitoring workstation 911, the process proceeds to step S1508 where, as discussed above, the remote monitoring workstation 911 resets the values corresponding to the first type of information stored in the database 913 to clear any conditions that may have been recorded leading up to the sending of the information. Once the database 913 values are reset, the process ends.

35 [0067] Figure 15B is a class diagram illustrating one embodiment of the Device Monitor 1103 module of Figure 11 according to the present invention. The Device Monitor 1103 module is responsible for logging and maintaining information about the network devices. This Device Monitor 1103 module is also responsible for ensuring that the information is sent to the central monitoring workstation 945. If the information is not to be sent upon being collected, the Device Monitor 1103 module obtains and stores only certain types of information (e.g., occurrence of no toner, door open, jam, etc.) in the database 913. If the information is to be sent upon being collected, the Device Monitor 1103 module obtains other types of information, including, for example, less volatile status information. The Device Monitor 1103 module includes three classes, CDeviceStatusMonitorAndSendManager 1601, CDeviceStatusLogger 1603, and CSnmpResource 1607.

40 [0068] The CDeviceStatusMonitorAndSendManager 1601 class is responsible for obtaining the information from the

monitored devices and sending the information to the central monitoring workstation 945. The CDeviceStatusMonitorAndSendManager 1601 class uses the Data Transfer 1107 module discussed above to send the information to the central monitoring workstation 945.

[0069] The CDeviceStatusLogger 1603 class is responsible for logging and maintaining the information of the monitored devices. The CDeviceStatusLogger 1603 class obtains and stores information of the monitored devices in the database 913 using the ODBC Interface 1109 module discussed above. The CDeviceStatusLogger 1603 class includes the DevicePerStatus structure for storing the first type of information of the monitored devices in the database 913. In one embodiment of the present invention, only this first type of information is stored in the database.

[0070] The CSnmpResource 1607 class is responsible for providing the network management protocol (e.g., SNMP) which provides the capability for collecting the information from the monitored devices. The CSnmpResource 1607 class uses the SNMP++ DLL 1609 to implement the Simple Network Management Protocol to gather the information from the monitored devices.

[0071] Figure 16 is a collaboration diagram illustrating the interaction among the classes of the Device Monitor 1103 module to obtain and store the first type of information from the monitored devices. As discussed above, if the information collected from the monitored devices is not to be sent to the central monitoring workstation 945 upon collection, only this first type of information is collected and stored in the database 913. As shown in Figure 16, the process is initiated by the CDeviceStatusMonitorAndSendManager 1601 class invoking the logDevicePerStatus() method of the CDeviceStatusLogger 1603 class to initiate the collection and storing of the first type of information of the monitored devices. The CDeviceStatusLogger 1603 class then calls the getDevicePerStatus() method of the ODBC Interface 1109 module to obtain the latest information of the monitored devices, including the IP addresses of the devices, from the database 913. Next, the CDeviceStatusLogger 1603 class calls the setIPAddressOfAgent() method of the CSnmpResource 1607 class which, in turn, calls the set_address() method of the SNMP++ DLL 1609 to establish an IP address of a device from which the first type of information is to be collected. Next, the CDeviceStatusLogger 1603 class calls the getOctetStringValueForOID() method of the CSnmpResource 1607 class which, in turn, calls the get() method and the get_value() method of the SNMP++ DLL 1609 to obtain the latest information of the monitored devices via SNMP using the IP address of the device. Once the information has been returned, the CDeviceStatusLogger 1603 class calls the setDevicePerStatus() method of the ODBC Interface 1109 module to store the information of the monitored devices in the database 913 using the DevicePerStatus 1505 structure.

[0072] Figure 17 is a collaboration diagram illustrating the interaction among the classes of the Device Monitor 1103 module to obtain both a first type and a second type of information from the monitored devices and to reset the values corresponding to the first type of information stored in the database 913 for each monitored device. As discussed above, if the information collected from the monitored devices is to be sent to the central monitoring workstation 945 upon collection, both the first type of information and the second type of information is collected. As shown in Figure 17, the process is initiated by the CDeviceStatusMonitorAndSendManager 1601 class invoking the getNextDeviceStatus() method of the CDeviceStatusLogger 1603 class to initiate the collection of the information (both the first type, and the second type of information) of the monitored devices. Steps 2-8 in collecting both types of information are the same as steps 2-8 described above (i.e., the calls to getDevicePerStatus(), setIPAddressOfAgent(), set_address(), getOctetStringValueForOID(), get(), get_value(), and setDevicePerStatus()) in the context of Figure 16 for collecting only the first type of information. However, the call to the setDevicePerStatus() method of the ODBC Interface 1109 module in this case (i.e., step 8) is used to reset the values corresponding to the first type of information stored in the database.

[0073] After resetting the values in the database, the CDeviceStatusLogger 1603 class calls the getStringValueForOID() method of the CSnmpResource 1607 class which, in turn, calls the get() and get_printable_value() methods of the SNMP++ DLL 1609 module to obtain the second type of information from the monitored devices via SNMP commands.

[0074] Figure 18 is a collaboration diagram illustrating the interaction among the classes of the Device Monitor 1103 module to send both the first type and second type of information of the monitored devices to the central monitoring workstation 945. As shown in Figure 18, the process is initiated by the CDeviceStatusMonitorAndSendManager 1601 class invoking the getNextDeviceStatus() method of the CDeviceStatusLogger 1603 class to obtain the information of the monitored devices as discussed above in the context of Figure 17. Next, the CDeviceStatusMonitorAndSendManager 1601 class calls the startSend() and dataSend() methods of the Data Transfer 1107 module to initiate the sending of the information to the central monitoring workstation 945. Then, the CDeviceStatusMonitorAndSendManager 1601 class iteratively calls the getNextDeviceStatus() method of the CDeviceStatusLogger 1603 class to obtain information from a monitored device, followed by a call to the dataSend() method of the CDataTransfer 1405 class, shown in Figure 19C, of the Data Transfer 1107 module to send the information for a particular monitored device to the central monitoring workstation 945. Once information has been sent for all of the monitored devices, the CDeviceStatusMonitorAndSendManager 1601 class calls the endSend() method of the CDataTransfer 1405 class, shown in Figure 19C, of the Data Transfer 1107 module to complete the sending of the information.

[0075] Figure 19A is a flowchart providing an overview of the functions performed by the Data Transfer 1107 module in the context of the system diagram of Figure 9. This process focuses on the Data Transfer 1107 module responsible for sending the configuration and status information, and not the collection of the configuration and status information to be sent, as is described above in the context of the Device Information 1105 module and the Device Monitor 1103 module, respectively.

[0076] As shown in Figure 19A, the process begins with step S1901 where the system registry of the remote monitoring workstation 911 is populated with a source address and a destination address for e-mails transferring status information of the monitored devices. In this example, the source address will be the e-mail address of the remote monitoring workstation 911, and the destination address will be the e-mail address of the central monitoring workstation 945. Once the source and destination addresses have been populated in the system registry of the remote monitoring workstation 911, the transfer of information may begin.

[0077] The process then proceeds to step S1902 where a transfer of information commences. In step S1902, the remote monitoring workstation 911 accesses its system registry to obtain source and destination e-mail address information that will be used to populate header information for an e-mail message originating from the remote monitoring workstation 911 and having a destination of the central monitoring workstation 945.

[0078] Once the source and destination information has been obtained, the process proceeds to step S1903 where a communication link is established between the remote monitor workstation 911 and a Simple Mail Transfer Protocol (SMTP) Server 915. Once the communication link has been established, the process proceeds to step S1904, where the configuration or status information is sent as an e-mail message from the remote monitoring workstation 911 to the SMTP server 915 via the communication link. The SMTP server 915 will route the e-mail message to the appropriate recipient, in this case, the central monitoring workstation 945. In one embodiment of the present invention, the remote monitoring workstation 911 sends the configuration or status information as a Multipurpose Internet Mail Extensions (MIME) attachment to the Internet e-mail message. As discussed above, the configuration or status information, prior to sending it to the central monitoring workstation 945, is maintained in the database 913. Once the configuration or status information has been sent, the process proceeds to step S1905, where the remote monitoring workstation 911 will shut down the communication link between itself and the SMTP Server 915. Once the communication link has been shut down, the process ends.

[0079] Figure 19B is a flowchart describing in further detail the processing performed in sending configuration or status information as an attachment to an e-mail message (e.g., the process performed in step S1904 of Figure 19A) according to one embodiment of the present invention. As shown in Figure 19B, the sending the configuration or status information begins with step S1910 where the configuration or status information in the map structure is formatted to be sent. The structure of the map is such that either configuration or status information data may be stored in the map. Once the data to be sent from the remote monitoring workstation 911 to the central monitoring workstation 945 has been formatted, the process proceeds to step S1911 where the data is encrypted. The Data Transfer 1107 module is configured to allow for a level of encryption for a particular application to match that application's needs without impacting the interface of the Data Transfer 1107 module. Once the data is encrypted, the process proceeds to step S1912 where the data is encoded. Once the data has been encrypted and encoded, the process proceeds to step S1913 where the data is sent via the communication link described above, for example, as a MIME attachment to an e-mail message. Once the data has been sent, the process ends.

[0080] The transfer of configuration or status information from the remote monitoring workstation 911 to the central monitoring workstation 945 has been described in the context of Figures 19A and 19B as using a store-and-forward protocol. Using a store-and-forward approach, for example, SMTP and POP3, the e-mail message is sent via the SMTP server 915 to a mail server, for example mail server POP3 943, in Figure 9. The mail server POP3 will store the e-mail message until it is retrieved by the intended recipient, which, in the example discussed above, is the central monitoring workstation 945. When the central monitoring workstation 945 connects to the mail server POP3 943, the mail server POP3 will forward all messages it has stored that have the central monitoring workstation 945 as an intended recipient.

[0081] Figure 19C is a class diagram illustrating one embodiment of the Data Transfer 1107 module of Figure 11 according to the present invention. The Data Transfer 1107 module is responsible for formatting the configuration or status information collected from the monitored devices, and for sending that information as an attachment to an e-mail message from a remote monitoring workstation 911 to a central monitoring workstation 945, using, for example, SMTP. In one embodiment of the present invention, the monitoring information is sent as a MIME attachment to an e-mail message. The Data Transfer 1107 module is also responsible for encrypting the data and encoding the encrypted data using, for example, Base64 encoding before sending the data. The Data Transfer 1107 module includes six classes: CDataTransfer 1405, CSendManager 1903, CAbsEncrypter 1905, CNullEncrypter 1907, CBase64Encoder 1909, and CSmtlp 1911.

[0082] The CDataTransfer 1405 class provides the interface through which the functionality supported by the Data Transfer 1107 module is accessed. In one embodiment of the present invention, the CDataTransfer 1405 class includes four public methods through which all of the functionality of the Data Transfer 1107 module may be accessed. These

methods include a setDestination() method, a startSend() method, a dataSend() method, and an endSend() method. The setDestination() method is used to configure both a source and destination address for an e-mail from a remote monitoring workstation 911 to a central monitoring workstation 945. The startSend() method is used to initiate communications between the remote monitoring workstation 911 and a SMTP server 915. The dataSend() method is used to send the monitoring information as an e-mail message from the remote monitoring workstation 911 to the central monitoring workstation 945 via the SMTP server 915. The dataSend() method supports the sending of either configuration information or status information. The endSend() method is used to shut down the communication link after the configuration or status information has been sent. While the Data Transfer 1107 module includes significantly more capabilities, the complexities of these capabilities are hidden from the public interface.

[0083] Returning to Figure 19C, the CSendManager 1903 class includes classes that implement encrypting, encoding, and the management of a communication link between the remote monitoring workstation 911 and the SMTP server 915. CAbsEncrypter 1905 is an abstract class providing the flexibility to add new encryption methods by adding new derived classes of CAbsEncrypter 1905, such as, for example, CNullEncrypter 1907. This class structure provides the flexibility for an application to implement the desired level of encryption, or to change an encryption method without impacting the interface to the Data Transfer 1107 module.

[0084] The CBase64Encoder 1909 class provides base 64 encoding of the information before the information is sent. The CSmtip 1911 class is responsible for managing the communication link between the remote monitoring workstation 911 and the SMTP server 915. CSmtip 1911 makes use of the CSystemRegistry 1915 class for accessing the system registry to determine a source and destination e-mail address for the e-mail message header to be sent to the SMTP server 915. Furthermore, CSmtip 1911 makes use of the CSocket 1917 class available in the Microsoft foundation classes (MFC) for establishing and taking down the communication link between the remote monitoring workstation 911 and the SMTP server 915.

[0085] Figure 20A is a collaboration diagram illustrating the interaction among the classes of the Data Transfer 1107 module shown in Figure 19C when a remote monitoring workstation 911 initiates communication by establishing a link to the SMTP server 915. In Figure 20A, the user 2001 corresponds to either the Device Monitor 1103 module or the Device Information 1105 module. As shown in Figure 20A, the process is initiated by the user 2001 invoking the startSend() method of the CDataTransfer 1405 class. In calling the startSend() method, the user 2001 indicates which type of information (e.g., configuration information or status information) will be sent.

[0086] Figure 20B illustrates an exemplary MIME attachment including configuration information, as can be determined by the first line of the MIME attachment. Figure 20C, on the other hand, illustrates an exemplary MIME attachment including status information, as is also indicated by the first line of the MIME attachment. As described above, it is the startSend() method that ensures that the first line is appropriately populated. The exemplary MIME attachments shown in Figures 20B and 20C are neither encrypted nor encoded.

[0087] Returning to Figure 20A, once the user 2001 has requested the initiation of communications, the CDataTransfer 1405 class will call the startSend() method of the CSendManager 1903 class. The CSendManager 1903 class manages the establishment of a communication link between the remote monitoring workstation 911 and the SMTP server 915 through interactions with the CSmtip 1913 class, which, in turn, interacts with the CSocket 2013 class.

[0088] In order to use SMTP, the CSendManager 1903 class calls the createSocket() method of the CSmtip 1913 class to create a socket to the SMTP server 915 through which the SMTP commands will be sent. Next, the CSendManager 1903 class calls the connectSocket() method of the CSmtip 1913 class to connect to that socket of the SMTP server 915. In response, the CSmtip 1913 class will call the Connect() and Receive() methods of the CSocket 2013 class to connect to the socket.

[0089] Once the socket has been established and connected to, the CSendManager 1903 class calls the sendHelloCommand() method of the CSmtip 1913 class to send the HELO SMTP command to the SMTP server 915. In response the CSmtip 1913 class will call the Send() method of the CSocket 2013 class to send the command to the socket of the SMTP server 915, and subsequently call the Receive() method of the CSocket 2013 class to receive a response from the socket of the SMTP server 915. Using the same approach, the CSendManager 1903 class will send the MAIL, RCPT, and DATA SMTP commands to the SMTP server 915 by respectively calling the sendMailCommand(), sendRcptCommand(), and sendDataCommand() methods of the CSmtip 1913 class. In response to each of the calls, the CSmtip 1913 class will call the Send() and Receive() methods of the CSocket 2013 class to send the commands to, and receive a response from, the socket of the SMTP server 915, respectively.

[0090] As would be understood by those of ordinary skill in the SMTP art, the HELO SMTP command is used by a client, for example, the remote monitoring workstation 911, to identify itself to the SMTP server 915, the MAIL SMTP command is used to identify the originator of a mail message, the RCPT SMTP command is used to identify the recipient for a mail message, and the DATA SMTP command is used to send the contents of a mail message.

[0091] Next, the CSendManager 1903 class calls the sendMailHeader() method of the CSmtip 1913 class to send the mail header for the e-mail message. The CSmtip 1913 class then calls the Send() method of the CSocket 2013 class to send the header to the socket of the SMTP server 915. At this point, the information to be included in the e-

mail message can be sent through the socket of the SMTP server 915. The data corresponding to the appropriate information type of the monitored devices is sent by the CDataTransfer 1405 class calling the sendData() method of the CSendManager 1903 class.

[0092] Figure 21 is a collaboration diagram illustrating the interaction among the classes of the Data Transfer 1107 module shown in Figure 19C to send either status or configuration data through e-mail in response to the Device Monitor 1103 module or the Device Information 1105 module, respectively. In Figure 21, the user 2001 corresponds to either the Device Monitor 1103 module or the Device Information 1105 module. As shown in Figure 21, the process is initiated by the user 2001 calling the dataSend() method of the CDataTransfer 1405 class. The user 2001 sends the map containing the configuration or status information as a parameter of the dataSend() method. The dataSend() method provides a single interface, through which both the Device Monitor 1103 module and the Device Information 1105 module provide information to be sent to the Data Transfer 1107 module. In response, the CDataTransfer 1405 class calls the sendData() method of the CSendManager 1903 class to send the information to the SMTP server 915. In one embodiment of the present invention, each call to the sendData() method of CSendManager 1903 class sends one key/data value pair stored in the map to the socket (steps 8 and 9 shown in Figure 21 are illustrative of this approach). Prior to sending the data, however, the CSendManager 1903 class calls the encryptData() method of the CNullEncrypter 1907 class (or, as described above, another class derived from the CAbsEncrypter 1905 abstract class) to encrypt the data to be sent.

[0093] Next, the CSendManager 1903 class calls the encodeData() and getEncodedString() methods of the CBase64Encoder 1911 class to encode the encrypted data. To send the encrypted and encoded data, the CSendManager 1903 class calls the sendData() method of the CSmtip 1913 class, which, in turn, calls the Send() method of the CSocket 2013 class.

[0094] Figure 22 is a collaboration diagram illustrating the interaction among the classes of the Data Transfer 1107 module shown in Figure 19C when the sending of the configuration or status information through e-mail has been completed. In Figure 22, the user 2001 corresponds to either the Device Monitor 1103 module or the Device Information 1105 module. As shown in Figure 22, the process is initiated by the user 2001 calling the endSend() method of the CDataTransfer 1405 class. In response, the CDataTransfer 1405 class calls the sendData() method of the CSendManager 1903 class to send data indicating the end of the data to be sent. While it is not shown in Figure 22, the data sent will be encrypted and encoded following the same approach as that described above in the context of Figure 21. Next, the CDataTransfer 1405 class calls the endSend() method of the CSendManager 1903 class to complete the sending. The CSendManager 1903 class completes the sending of data by first calling the endOfData() and getEncodedString() methods of the CBase64Encoder 1911 class to obtain the last encoded information to be sent. Next, the CSendManager 1903 class calls the sendData() method of the CSmtip 1913 class to send the last encoded encoded string containing data. To send the last encoded string, the CSmtip 1913 class calls the Send() method of the CSocket 2013 class.

[0095] Next, the CSendManager 1903 class calls the sendEndOfMail() method of the CSmtip 1913 class to send the end of the mail data. In turn, the CSmtip 1913 class calls the Send() method of the CSocket 2013 class to send the end of the mail data through the socket of the SMTP server 915 followed by a call to the Receive() method of the CSocket 2013 class to obtain a response from the socket. Next, the CSendManager 1903 class calls the sendQuitCommand() method of the CSmtip 1913 class to send the QUIT SMTP command to the socket of the SMTP server 915 to terminate the e-mail session between the remote monitoring workstation 911 and the SMTP server 915. In response, the CSmtip 1913 class calls the Send() and Receive() methods of the CSocket 2013 class to send the QUIT command and obtain a response from the socket.

[0096] Figure 23 is a collaboration diagram illustrating the interaction among the classes of the Data Transfer 1107 module shown in Figure 19C to set up the system registry of the remote monitoring workstation 911 for sending device configuration and status information through e-mail to the central monitoring workstation 945. As shown in Figure 23, the process is initiated by a call to the setDestination() method of the CDataTransfer 1405 class by the Interface 1101. In response, the CDataTransfer 1405 class calls the setDestination() method of the CSendManager 1403 class, which, in turn, calls the setDestination() method of the CSmtip 1913 class. To store the information in the system registry, the CSmtip 1913 class calls the setSMTPServer(), setFromAddr(), and setRcptAdd() methods of the CSystemRegistry 1915 class to store the SMTP Server, the from address, and the recipient address to be used in sending configuration or status information to the central monitoring workstation 945, respectively, in the system registry of the remote monitoring workstation 911.

[0097] Figure 24 is a class diagram illustrating one embodiment of the ODBC Interface 1109 module of Figure 11 according to the present invention. The ODBC Interface 1109 module is responsible for interfacing with the database 913 that maintains the information pertaining to the SNMP devices being monitored by a particular remote monitoring workstation 911. In this embodiment, the database is registered as an ODBC database, and, therefore, the database 913 has available the appropriate supporting ODBC drivers. The ODBC Interface 1109 module includes five classes: CSendODBCInterface 2401, CDeviceInformationData 2403, CDeviceDatabase 2407, CDevicePersistentStatus 2411,

and CDevicePerDatabase 2413. The CSendODBCInterface 2401 class provides the interface through which the functionality supported by the ODBC Interface 1109 module is accessed.

[0098] The CDeviceInformationData 2403 class provides methods for obtaining and storing configuration information of the monitored devices in the database 913. The CDeviceDatabase 2407 class provides an interface between the CDeviceInformationData 2403 class and the actual database 913 that contains the configuration information. The CDeviceInformationData 2403 class uses the DeviceInfo structure to store the configuration information in the database 913.

[0099] The CDevicePersistentStatus 2411 class provides methods for obtaining and storing the first type of information of the monitored devices in the database. The CDevicePerDatabase 2407 provides an interface between the CDevicePersistentStatus 2411 class and the actual database that contains the first type of information. The CDevicePersistentStatus 2411 class uses the DevicePerStatus structure to store the first type of information in the database 913.

[0100] Both the CDeviceDatabase 2407 class and the CDevicePerDatabase 2413 class are derived from the CRecordset 2417 class available in the Microsoft Foundation Classes (MFC).

[0101] Obviously, numerous modifications and variations of the present invention are possible in light of the above teachings. It is therefore to be understood that within the scope of the appended claims, the invention may be practiced otherwise than as specifically described herein.

Claims

1. A method for sending configuration information corresponding to a network device from a remote monitor to a central monitor, comprising the steps of:

querying a digital repository by the remote monitor for the configuration information, the configuration information including a local unique identifier corresponding to the network device;
obtaining a globally unique identifier from the network device by the remote monitor using the local unique identifier to access the network device using a network management protocol; and
sending the configuration information and the globally unique identifier by the remote monitor to the central monitor as a message using a transfer protocol.

2. The method of Claim 1, wherein the local unique identifier comprises an Internet protocol address of the network device.

3. The method of Claim 1, wherein the globally unique identifier comprises a physical address of the network device.

4. The method of Claim 1, wherein the physical address comprises a media access control address of the network device.

5. The method of Claim 1, wherein the network management protocol comprises a simple network management protocol.

6. The method of Claim 1, wherein the transfer protocol comprises a store-and-forward protocol.

7. The method of Claim 1, wherein the transfer protocol comprises a simple mail transfer protocol.

8. The method of Claim 1, wherein the transfer protocol comprises a direct connection protocol.

9. The method of Claim 1, wherein the transfer protocol comprises at least one of a file transfer protocol and a hyper text transfer protocol.

10. The method of Claim 1, wherein the message comprises an Internet e-mail message.

11. The method of Claim 1, wherein:

the message comprises an Internet e-mail message, and
the configuration information is sent as an attachment to the message.

12. The method of Claim 1, wherein:

the message comprises an Internet e-mail message, and
the configuration information is sent as a multipurpose Internet mail extensions attachment to the message.

13. The method of Claim 1, further comprising the step of:

encrypting the configuration information by the remote monitor prior to the sending step.

14. The method of Claim 13, further comprising the step of:

encoding the configuration information by the remote monitor after the encrypting step and prior to the sending step.

15. The method of Claim 14, wherein the encoding step comprises encoding the configuration information using a base 64 encoding.

16. The method of Claim 1, further comprising the step of:

encoding the configuration information by the remote monitor prior to the sending step.

17. The method of Claim 16, wherein the encoding step comprises encoding the configuration information using a base 64 encoding.

18. A computer-implemented system for sending configuration information corresponding to a network device from a remote monitor to a central monitor, comprising:

a processor; and
a computer readable medium encoded with processor readable instructions that when executed by the processor implement
a database management mechanism configured to query a database for the configuration information, the configuration information including a local unique identifier corresponding to the network device,
a device management mechanism configured to obtain a globally unique identifier from the network device using the local unique identifier to access the network device using a network management protocol, and
a sending mechanism configured to send the configuration information and the globally unique identifier from the remote monitor to the central monitor as a message using a transfer protocol.

19. The system of Claim 18, wherein the local unique identifier comprises an Internet protocol address of the network device.

20. The system of Claim 18, wherein the globally unique identifier comprises a physical address of the network device.

21. The system of Claim 18, wherein the globally unique identifier comprises a media access control address of the network device.

22. The system of Claim 18, wherein the network management protocol comprises a simple network management protocol.

23. The system of Claim 18, wherein the transfer protocol comprises a store-and-forward protocol.

24. The system of Claim 18, wherein the transfer protocol comprises a simple mail transfer protocol.

25. The system of Claim 18, wherein the transfer protocol comprises a direct connection protocol.

26. The system of Claim 18, wherein the transfer protocol comprises at least one of a file transfer protocol and a hyper text transfer protocol.

27. The system of Claim 18, wherein the message comprises an Internet e-mail message.

28. The system of Claim 18, wherein:

the message comprises an Internet e-mail message, and
the configuration information is sent as an attachment to the message.

29. The system of Claim 18, wherein:

the message comprises an Internet e-mail message, and
the configuration information is sent as a multipurpose Internet mail extensions attachment to the message.

30. The system of Claim 18, wherein the computer readable medium is further encoded with processor readable instructions that when executed by the processor further implements
an encryption mechanism configured to encrypt the configuration information prior to sending the configuration information.

31. The system of Claim 30, wherein the computer readable medium is further encoded with processor readable instructions that when executed by the processor further implements
an encoding mechanism configured to encode the configuration information after the encryption mechanism encrypts the configuration information and prior to sending the configuration information.

32. The system of Claim 31, wherein the encoding mechanism is further configured to encode the configuration information using a base 64 encoding.

33. The system of Claim 18, wherein the computer readable medium is further encoded with processor readable instructions that when executed by the processor further implements
an encoding mechanism configured to encode the configuration information prior to sending the configuration information.

34. The system of Claim 33, wherein the encoding mechanism is further configured to encode the configuration information using a base 64 encoding.

35. A computer program product, comprising:

a computer storage medium; and
a computer program code mechanism embedded in the computer storage medium for causing a computer to send configuration information corresponding to a network device from a remote monitor to a central monitor, the computer program code mechanism having
a first computer code device configured to query a database for the configuration information, the configuration information including a local unique identifier corresponding to the network device,
a second computer code device configured to obtain a globally unique identifier from the network device using the local unique identifier to access the network device using a network management protocol, and
a third computer code device configured to send the configuration information and the globally unique identifier from the remote monitor to the central monitor as a message using a transfer protocol.

36. The computer program product of Claim 35, wherein the local unique identifier comprises an Internet protocol address of the network device.

37. The computer program product of Claim 35, wherein the globally unique identifier comprises a physical address of the network device.

38. The computer program product of Claim 35, wherein the globally unique identifier comprises a media access control address of the network device.

39. The computer program product of Claim 35, wherein the network management protocol comprises a simple network management protocol.

40. The computer program product of Claim 35, wherein the transfer protocol comprises a store-and-forward protocol.

41. The computer program product of Claim 35, wherein the transfer protocol comprises a simple mail transfer protocol.

42. The computer program product of Claim 35, wherein the transfer protocol comprises a direct connection protocol.

5 43. The computer program product of Claim 35, wherein the transfer protocol comprises at least one of a file transfer protocol and a hyper text transfer protocol.

44. The computer program product of Claim 35, wherein the message comprises an Internet e-mail message.

10 45. The computer program product of Claim 35, wherein:

the message comprises an Internet e-mail message, and
the configuration information is sent as an attachment to the message.

15 46. The computer program product of Claim 35, wherein:

the message comprises an Internet e-mail message, and
the configuration information is sent as a multipurpose Internet mail extensions attachment to the message.

20 47. The computer program product of Claim 35, wherein the computer program code mechanism further having
a fourth computer code device configured to encrypt the configuration information prior to sending the configuration information.

25 48. The computer program product of Claim 47, wherein the computer program code mechanism further having
a fifth computer code device configured to encode the configuration information after the fourth computer code device encrypts the configuration information and prior to sending the configuration information.

30 49. The computer program product of Claim 48, wherein the fifth computer code device is further configured to encode
the configuration information using a base 64 encoding.

50. The computer program product of Claim 35, wherein the computer program code mechanism further having
a fourth computer code device configured to encode the configuration information prior to sending the configuration information.

35 51. The computer program product of Claim 50, wherein the fourth computer code device is further configured to
encode the configuration information using a base 64 encoding.

40

45

50

55

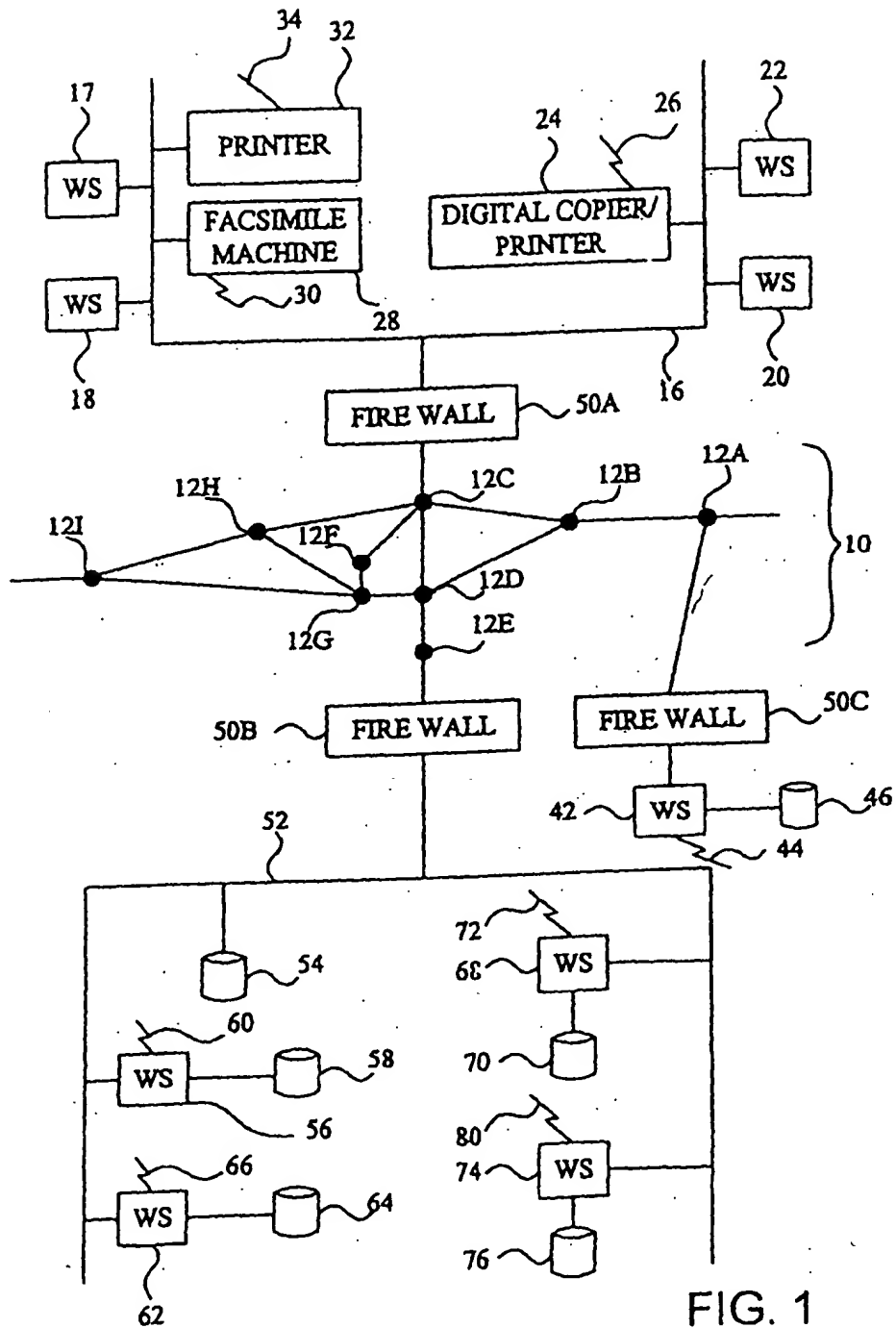


FIG. 1

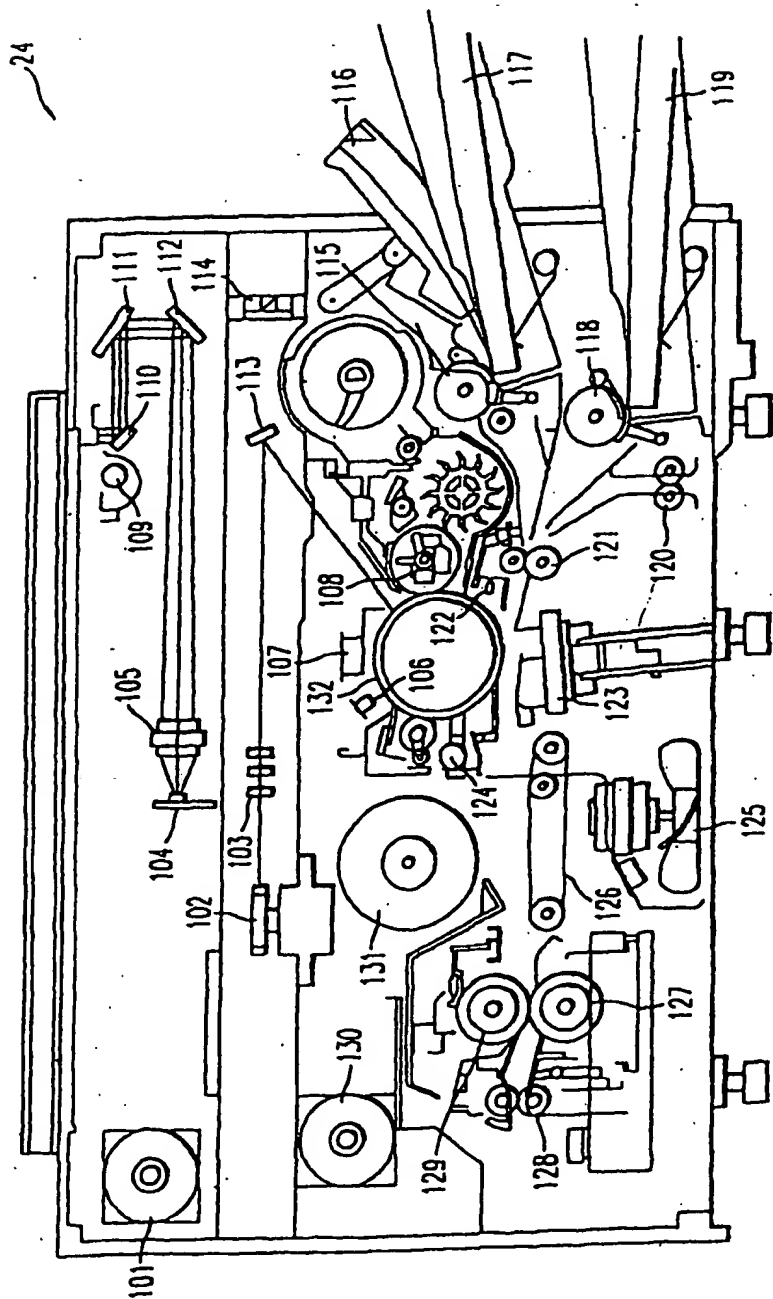
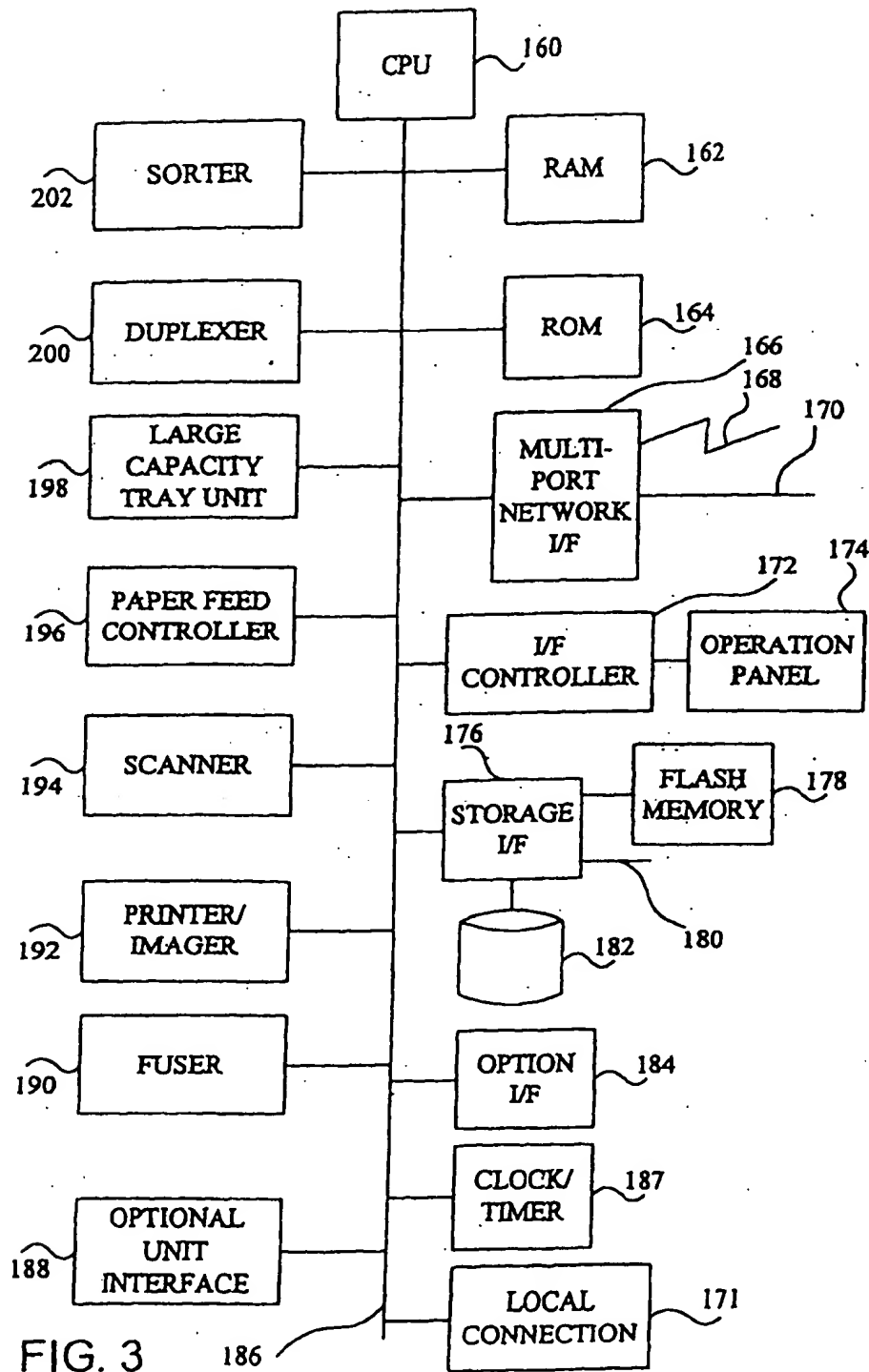


FIG. 2



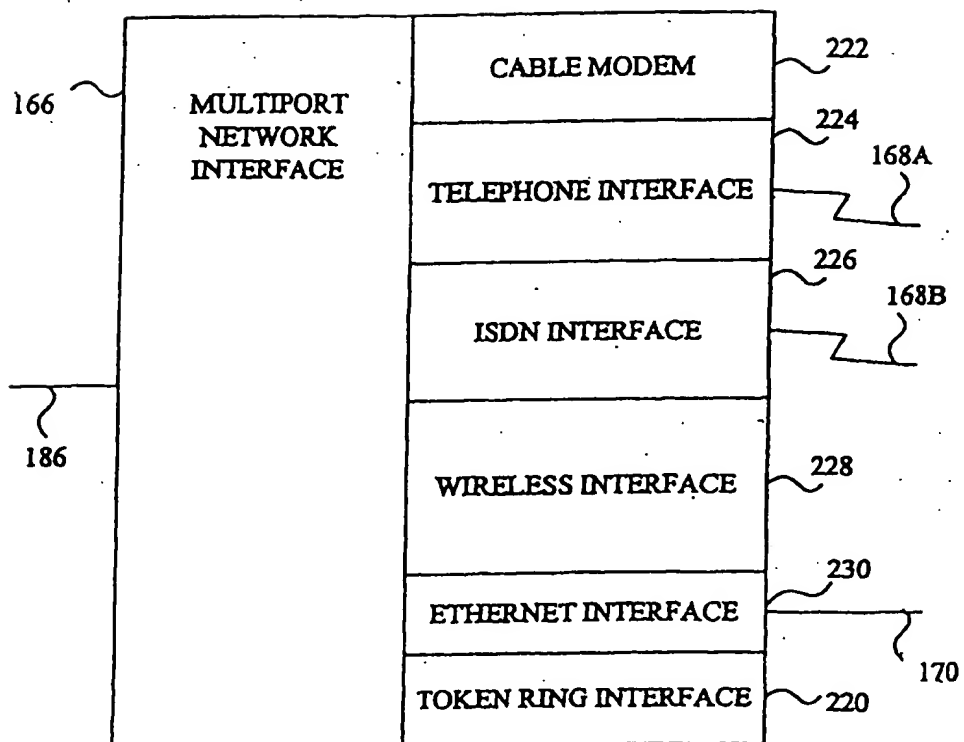
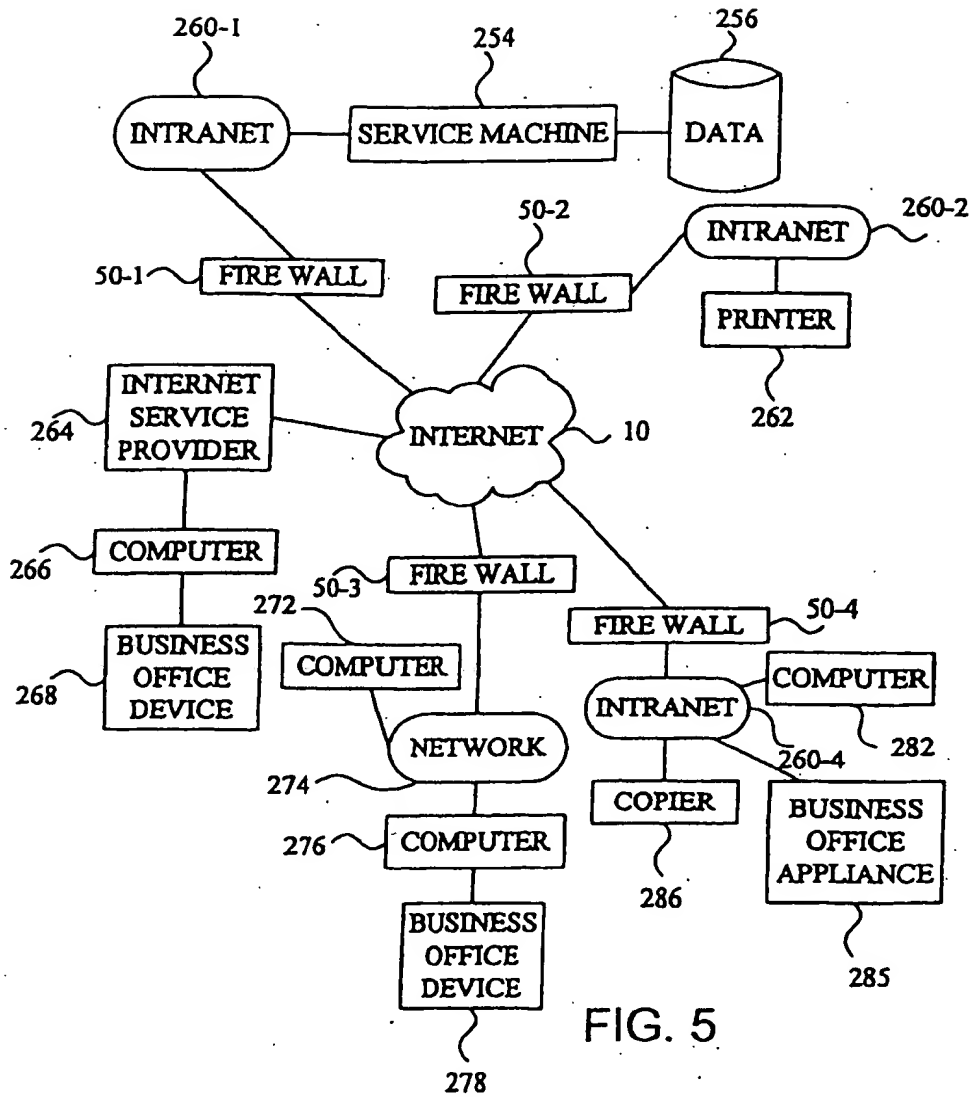


FIG. 4



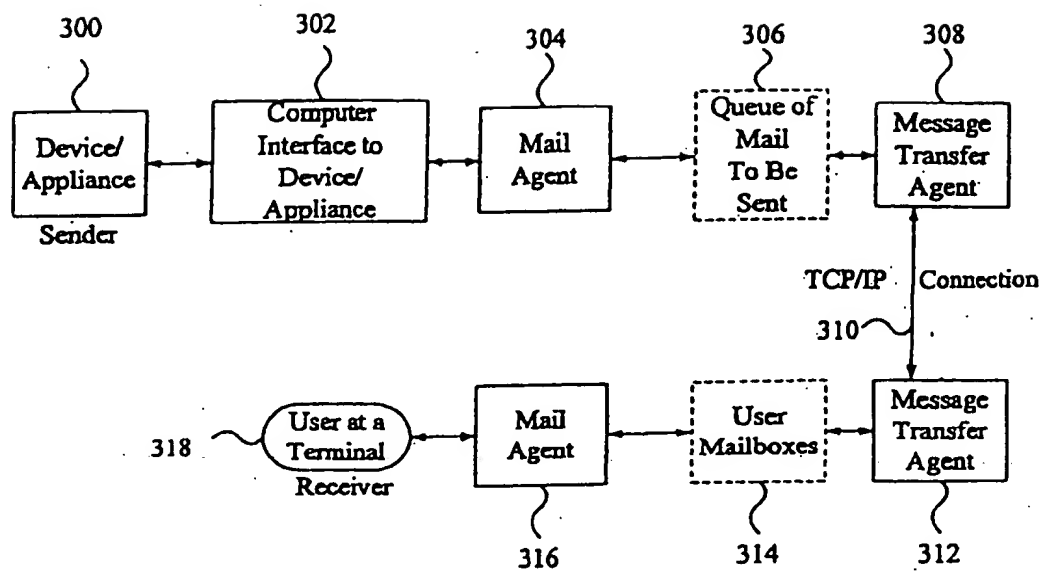


FIG. 6A

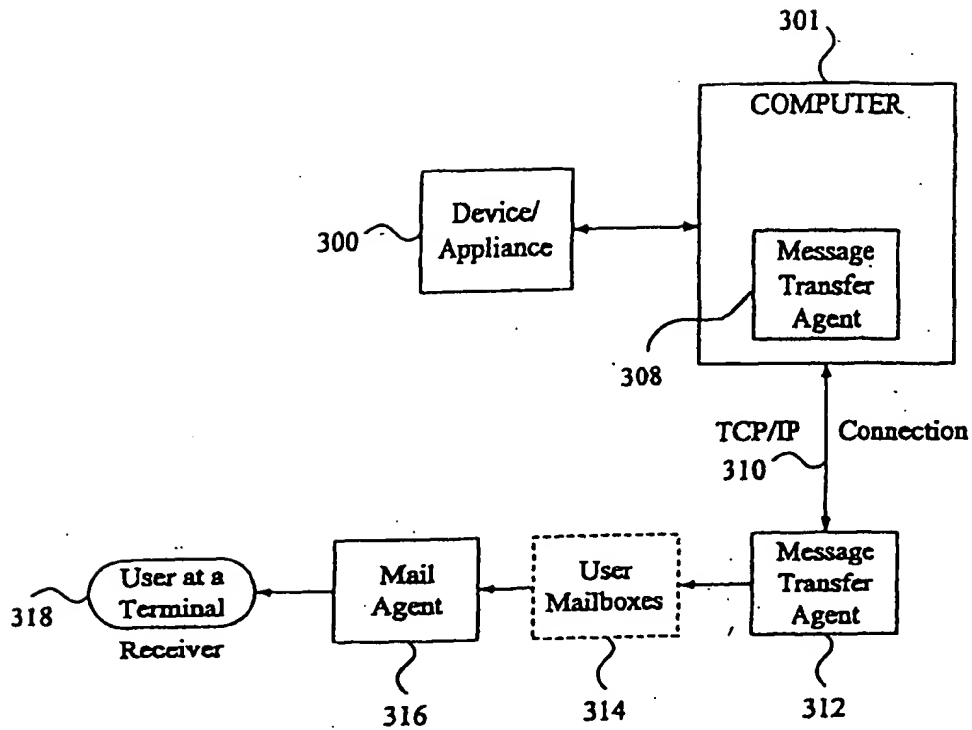


FIG. 6B

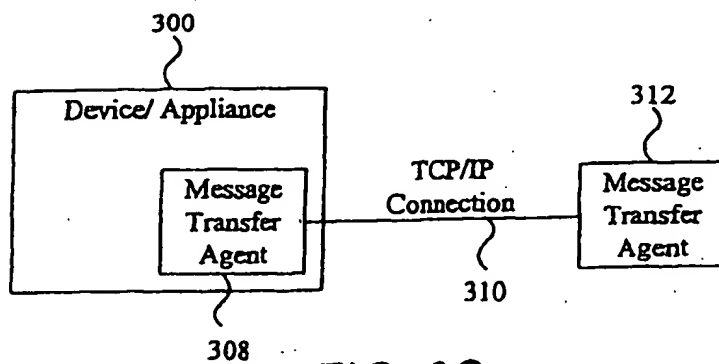


FIG. 6C

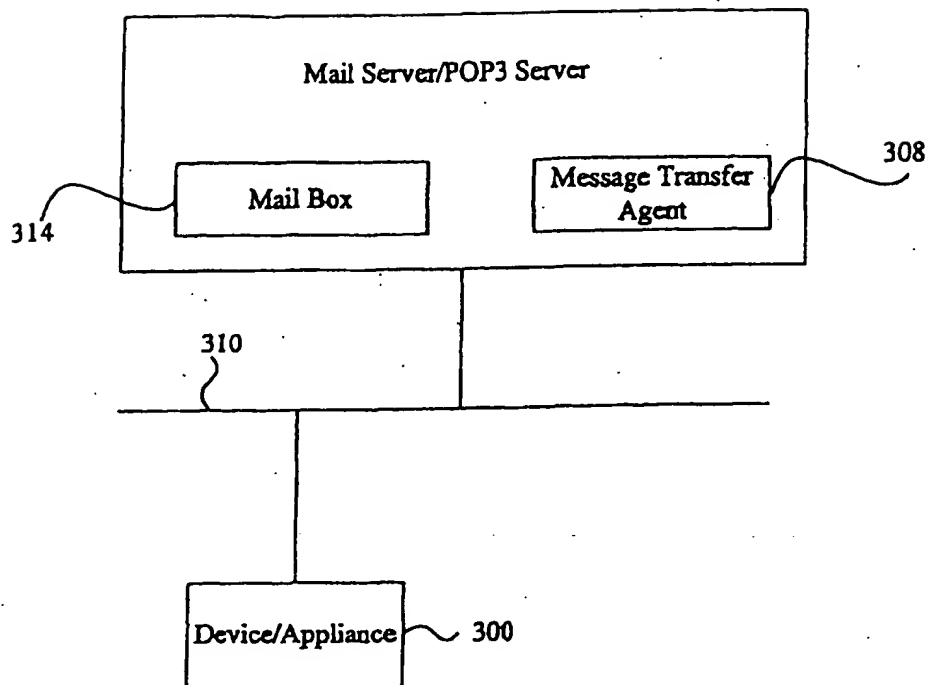


Fig. 6D

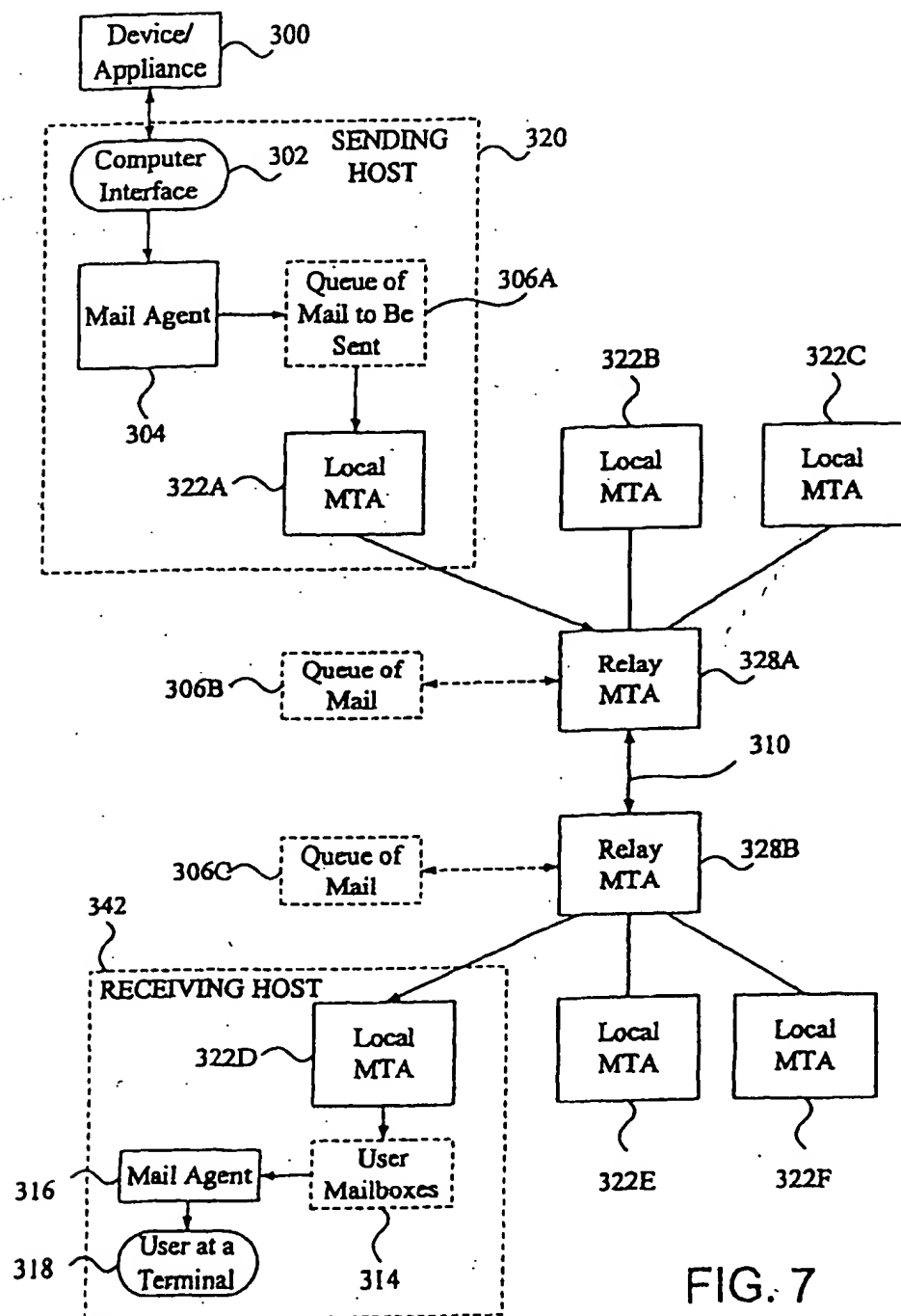


FIG. 7

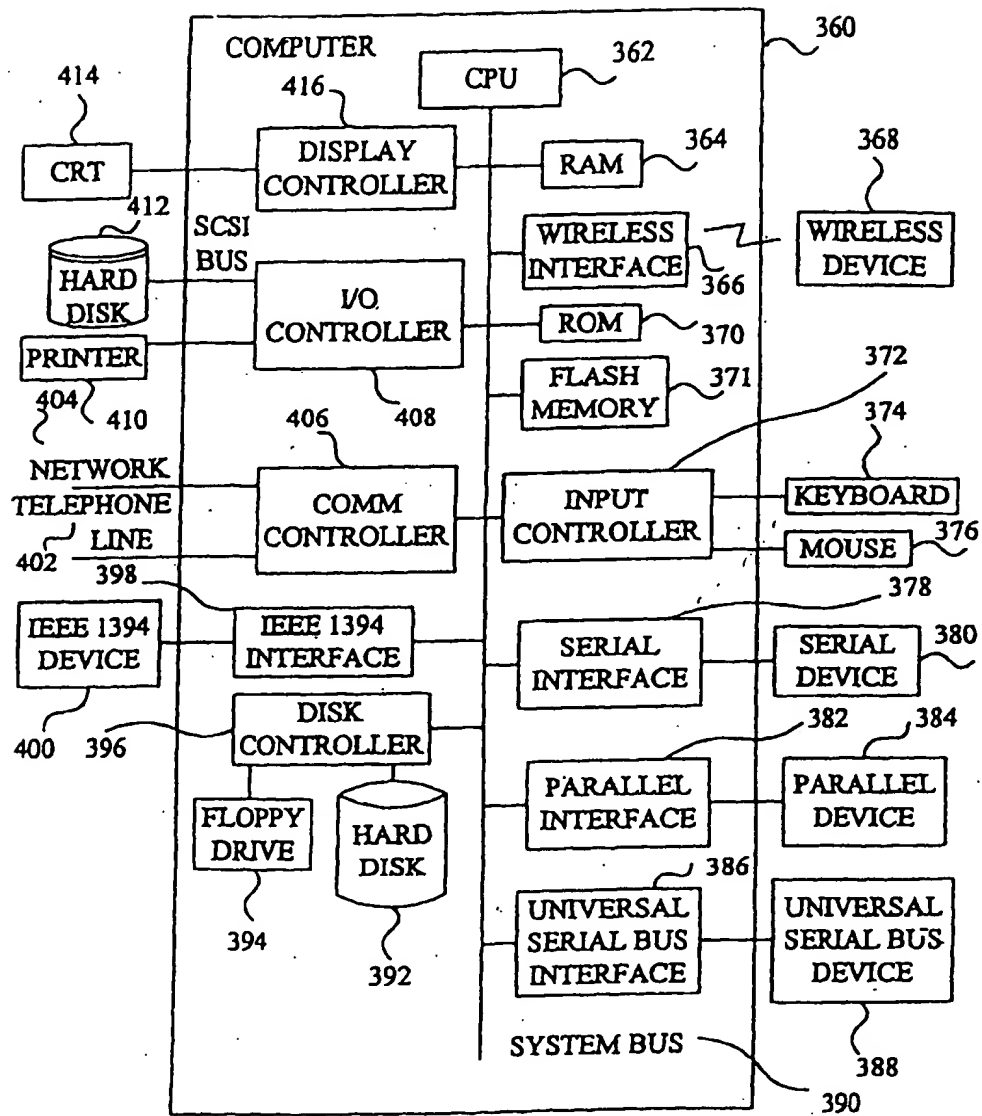


FIG. 8

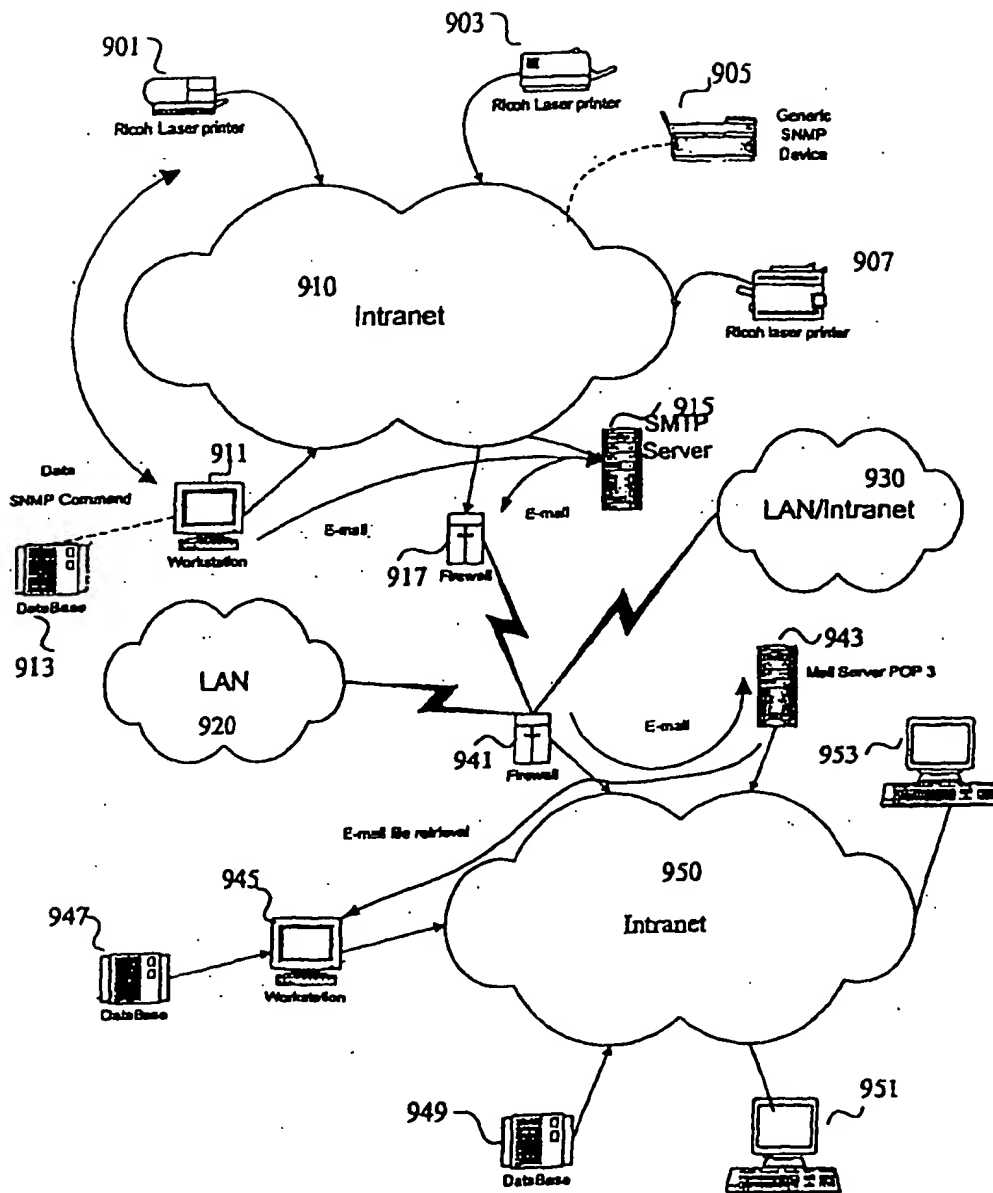


Figure 9

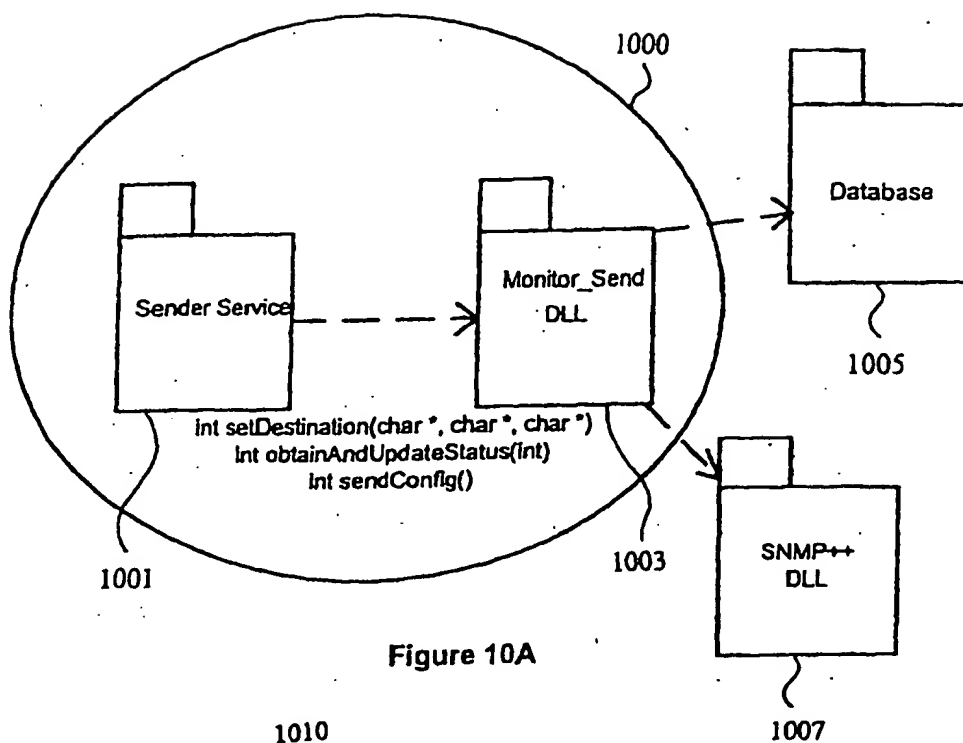


Figure 10A

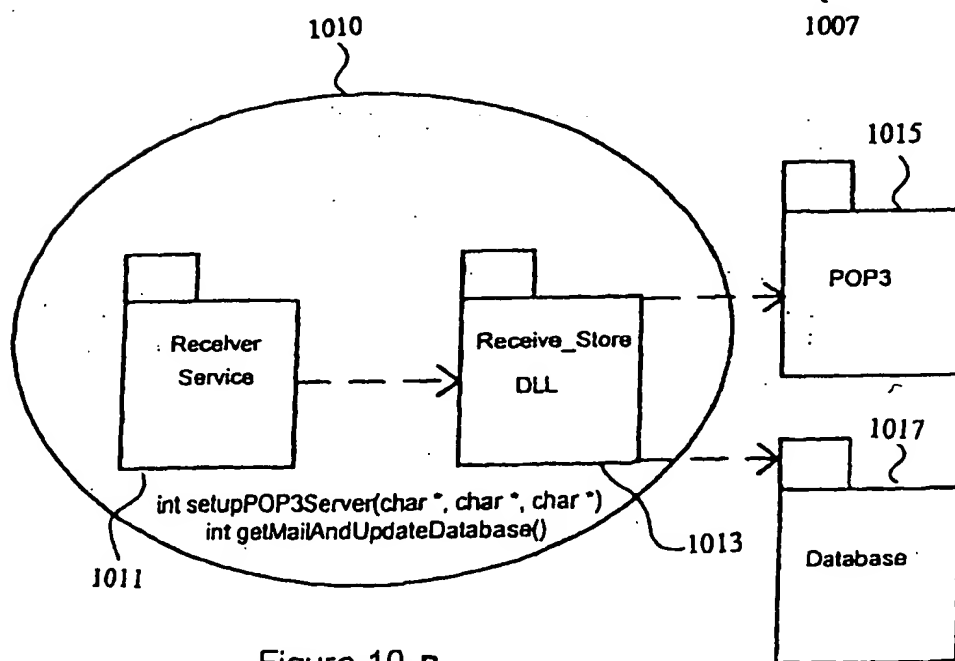


Figure 10 B

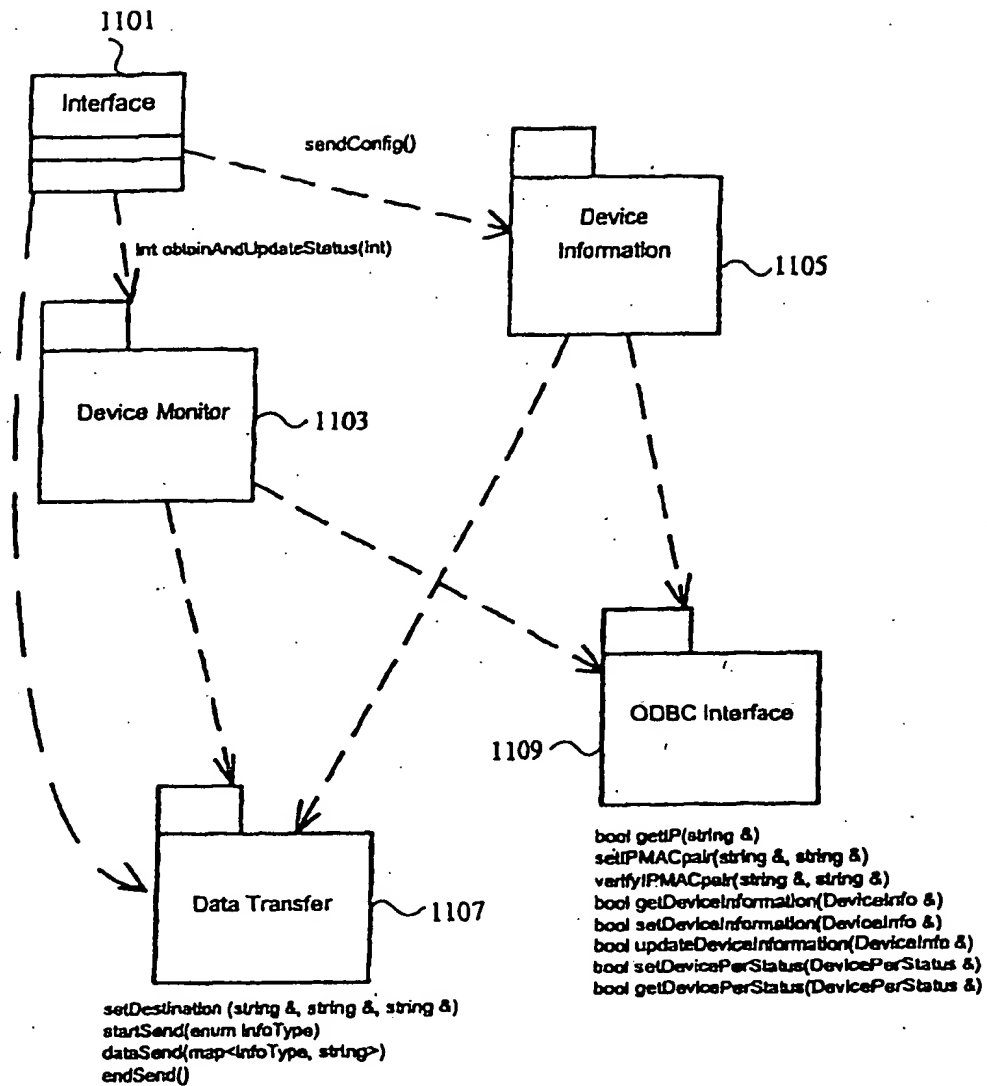


Figure 11

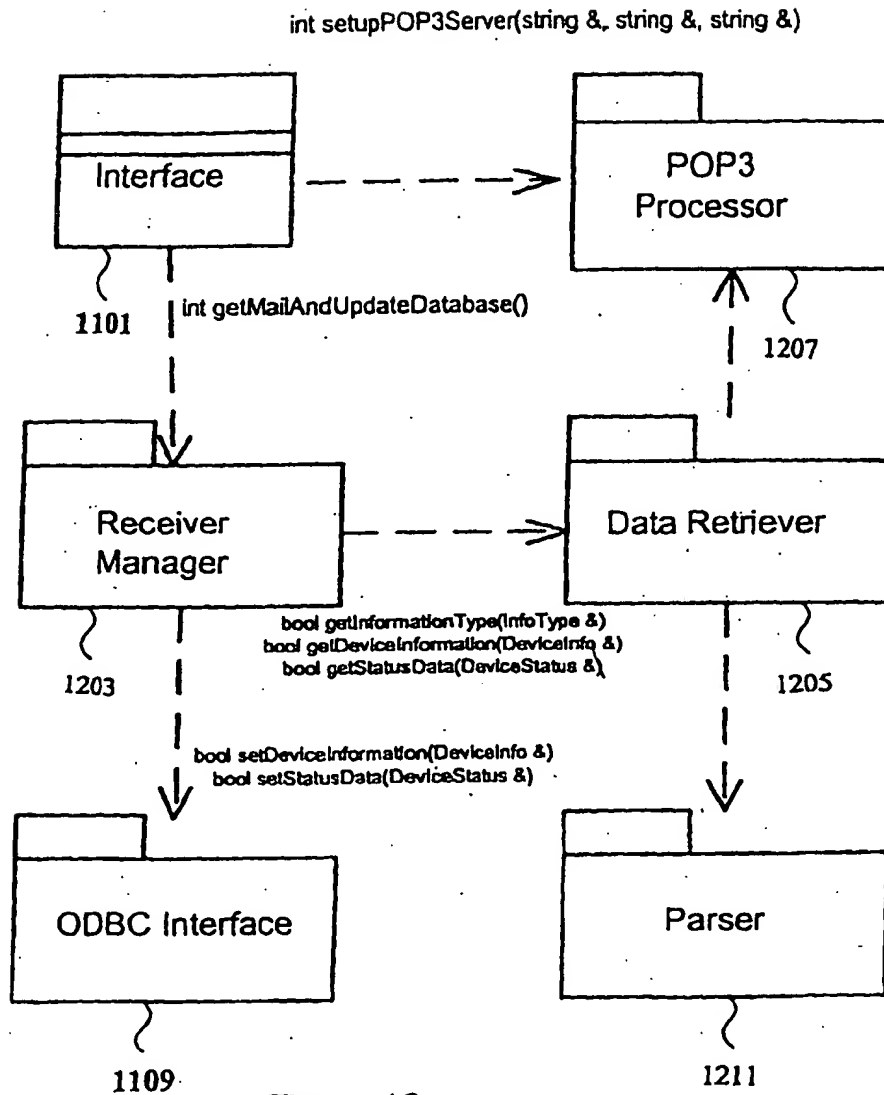
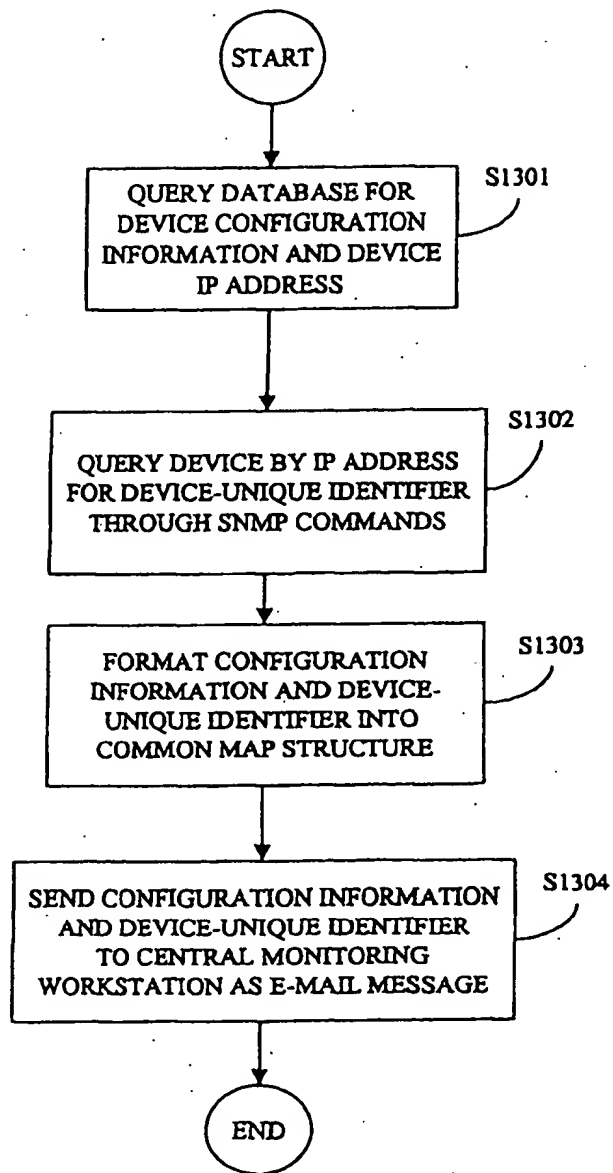
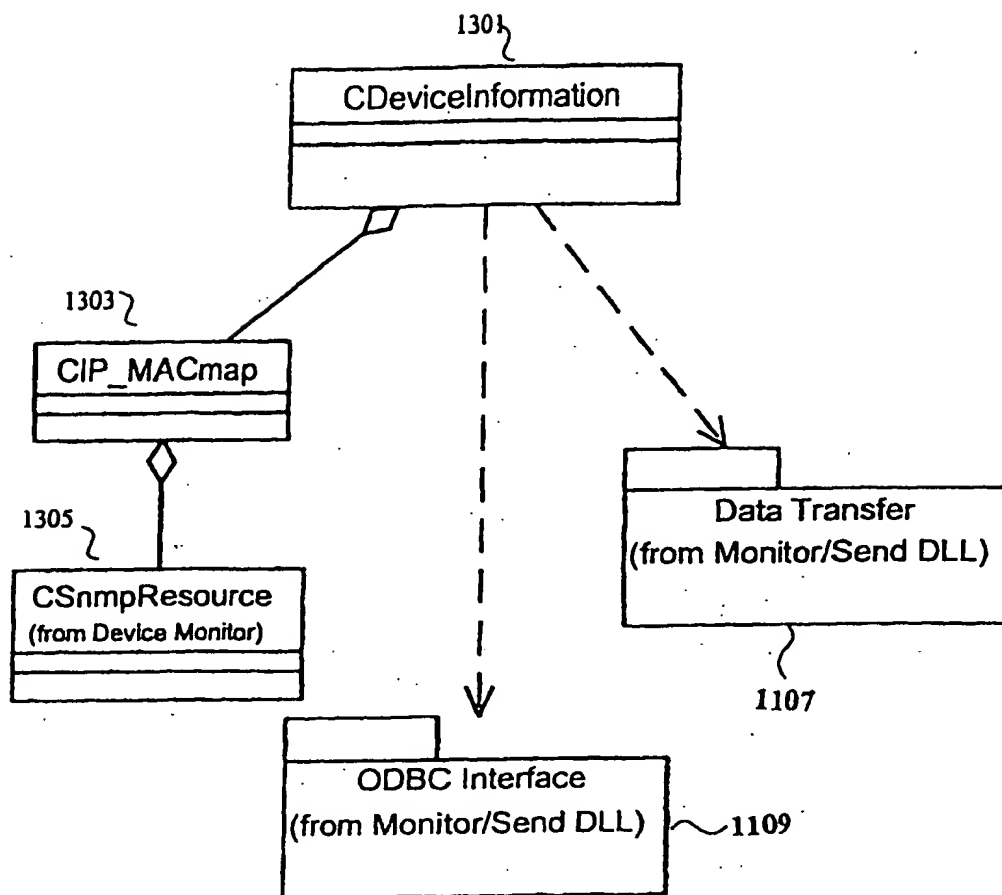


Figure 12

FIGURE 13A





Device Information package

Figure 13B

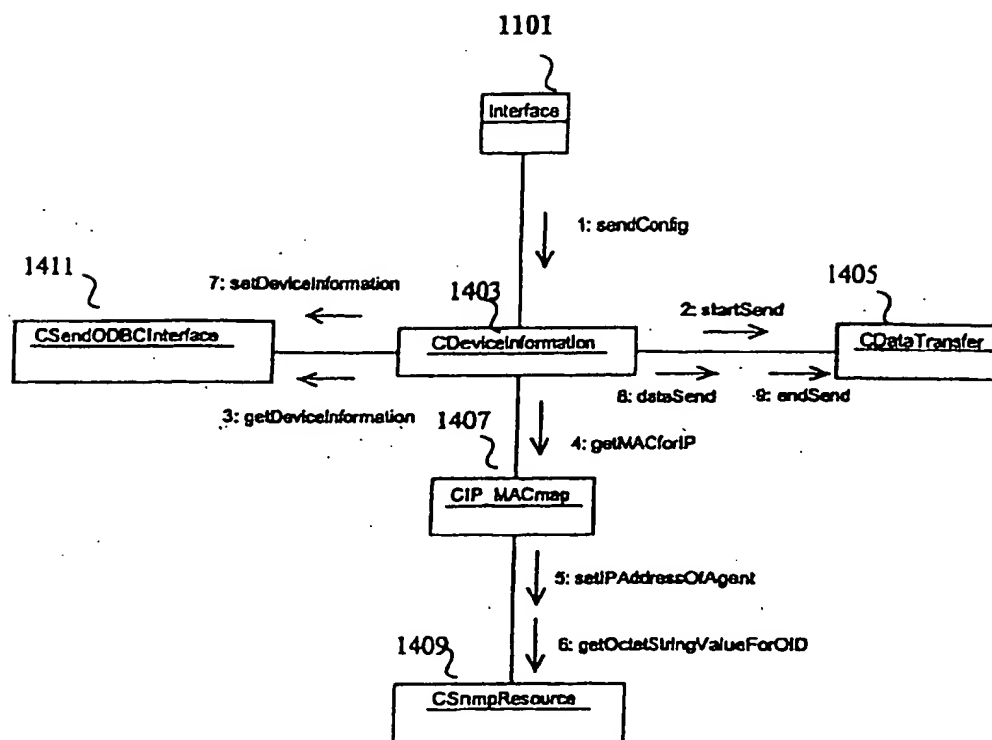
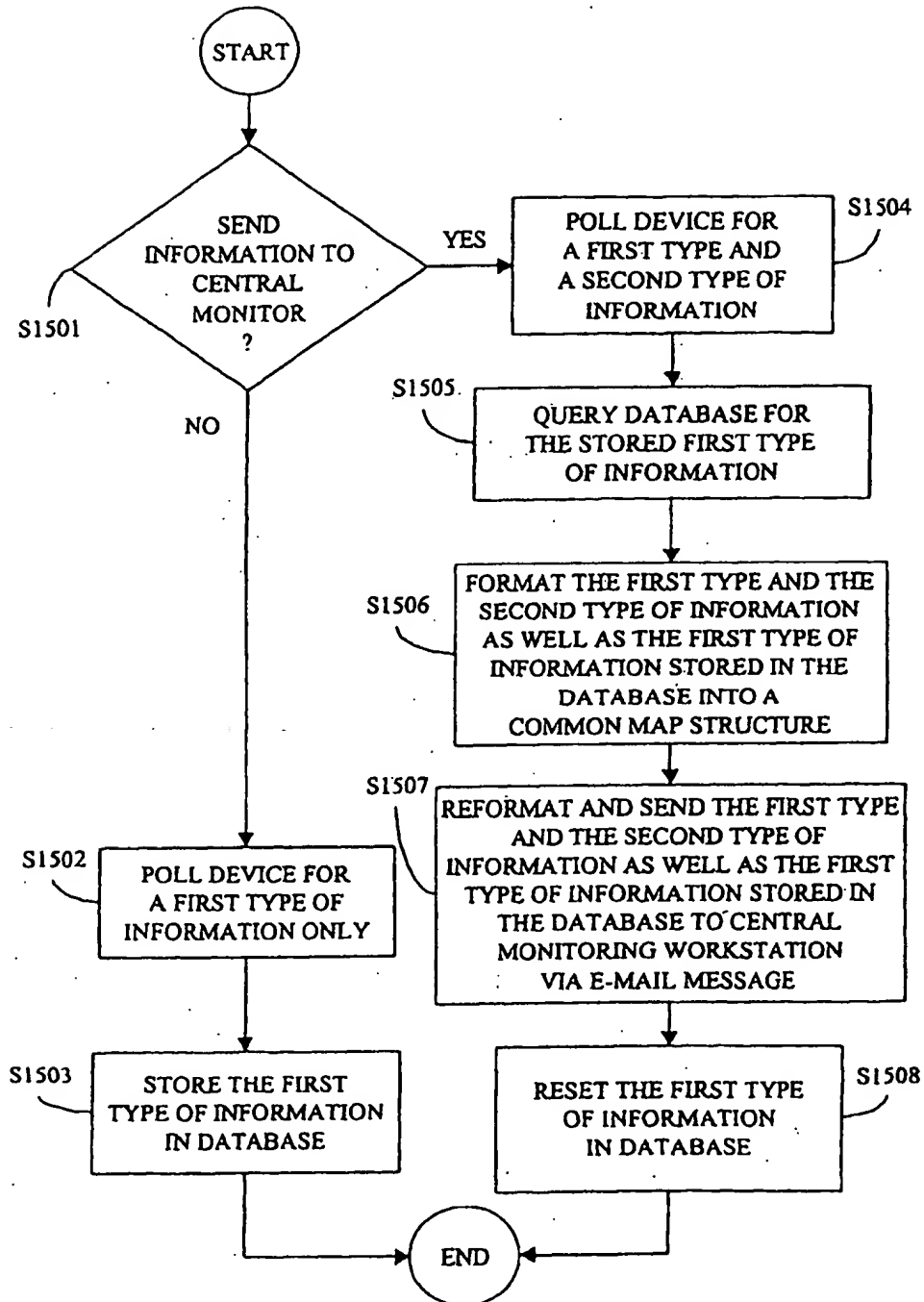
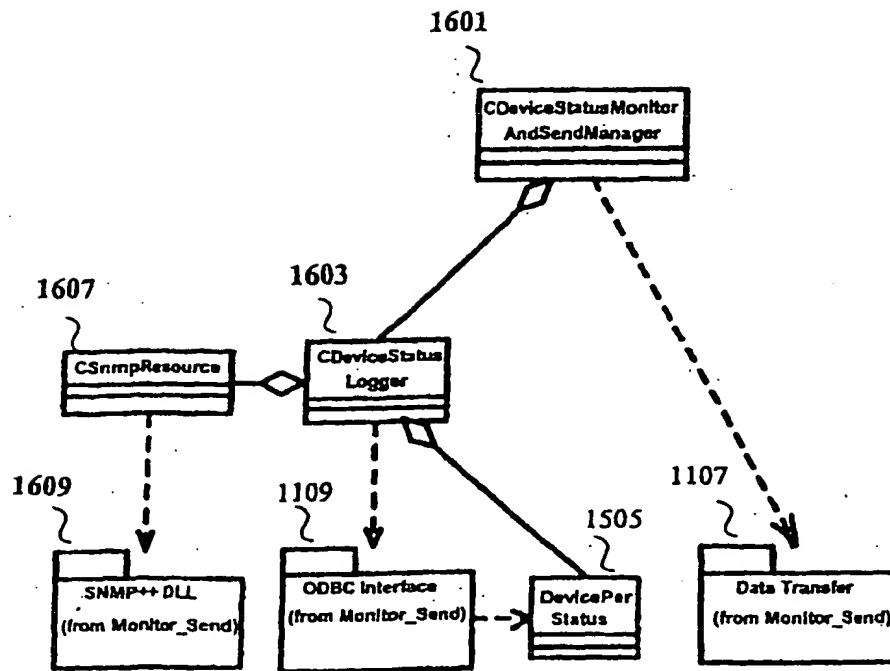


Figure 14

FIGURE 15A





Device Monitor Package Class Structure

Figure 15 B

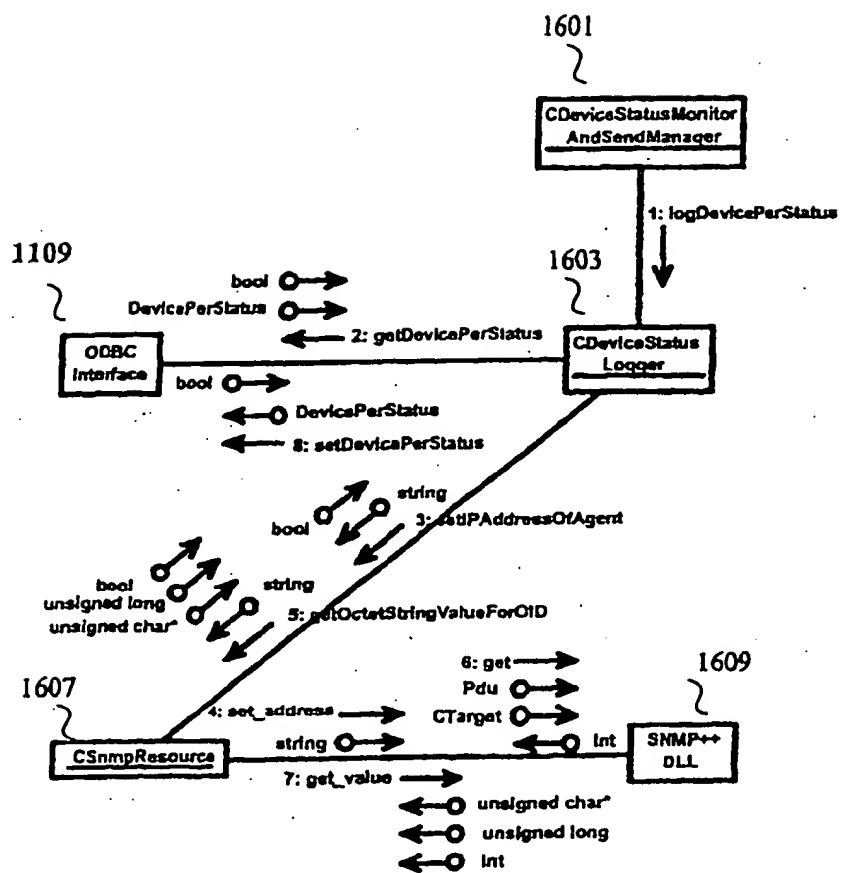


Figure 16

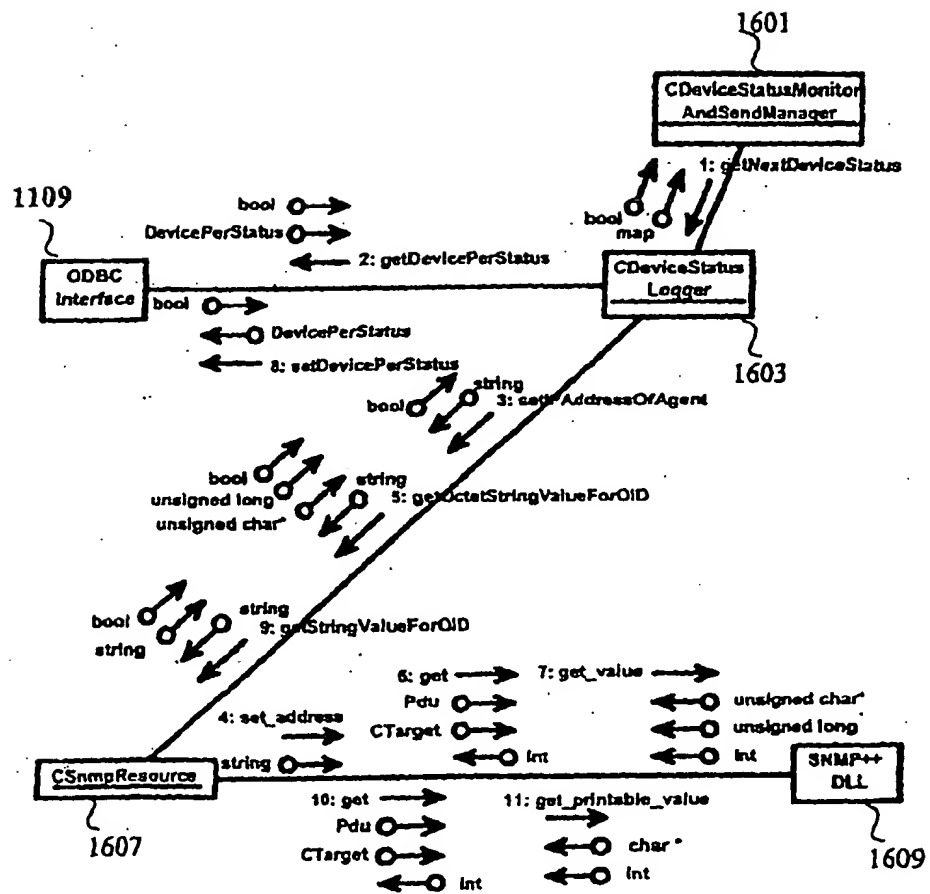


Figure 17

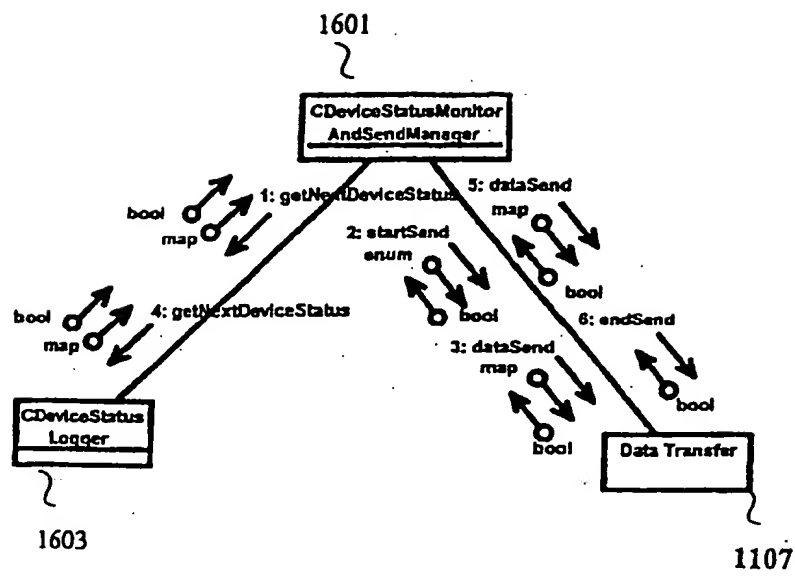


Figure 18

FIGURE 19A

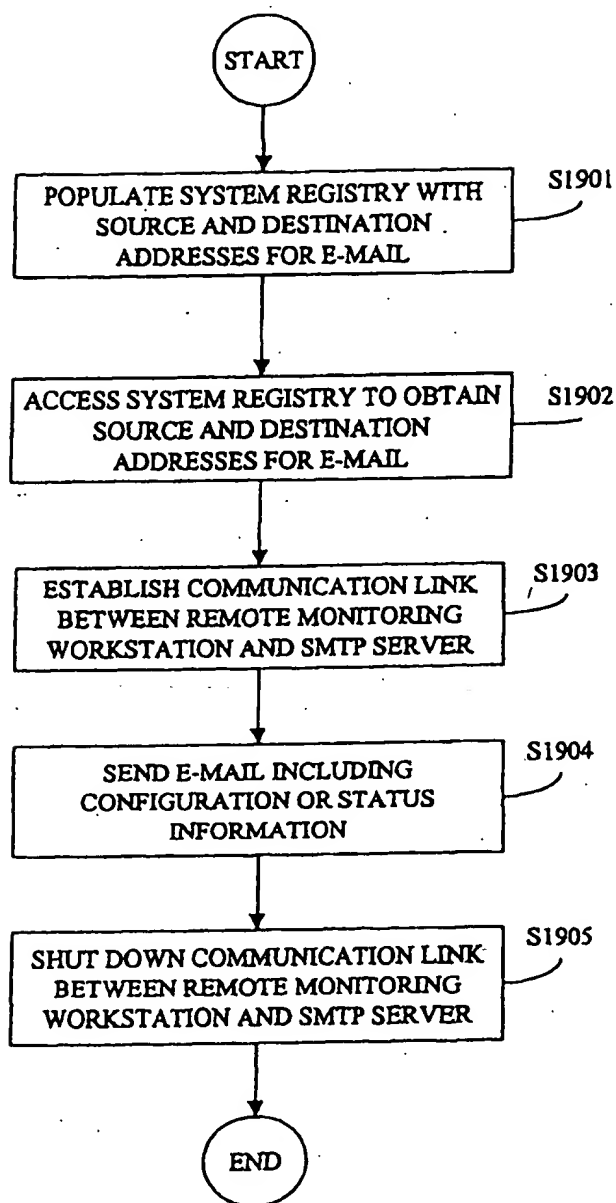
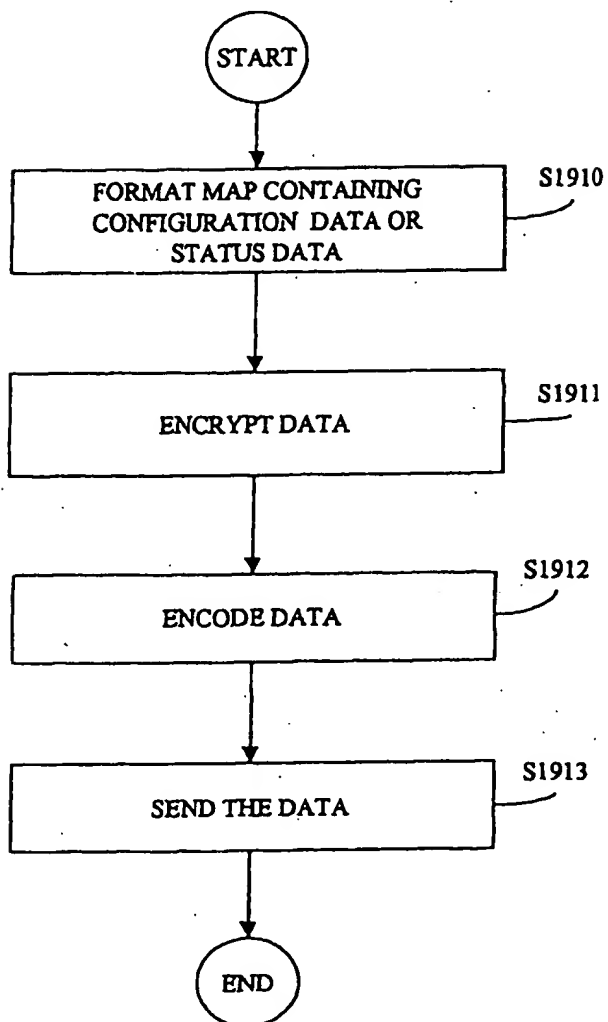
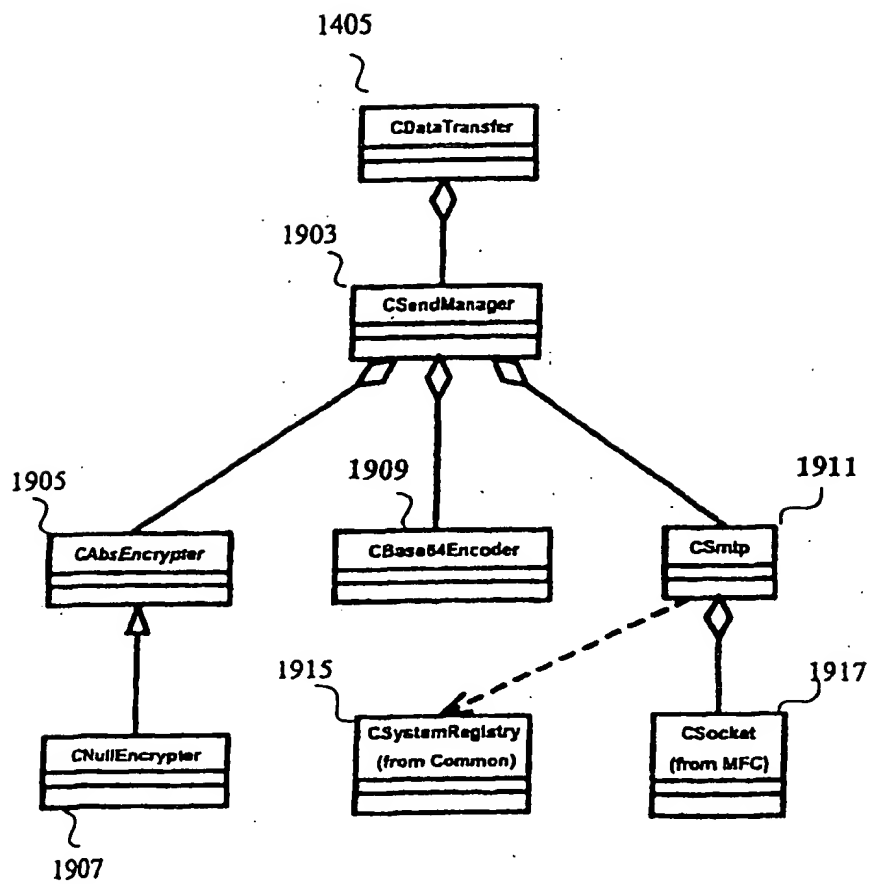


FIGURE 19B





Data Transfer Package

Figure 19C

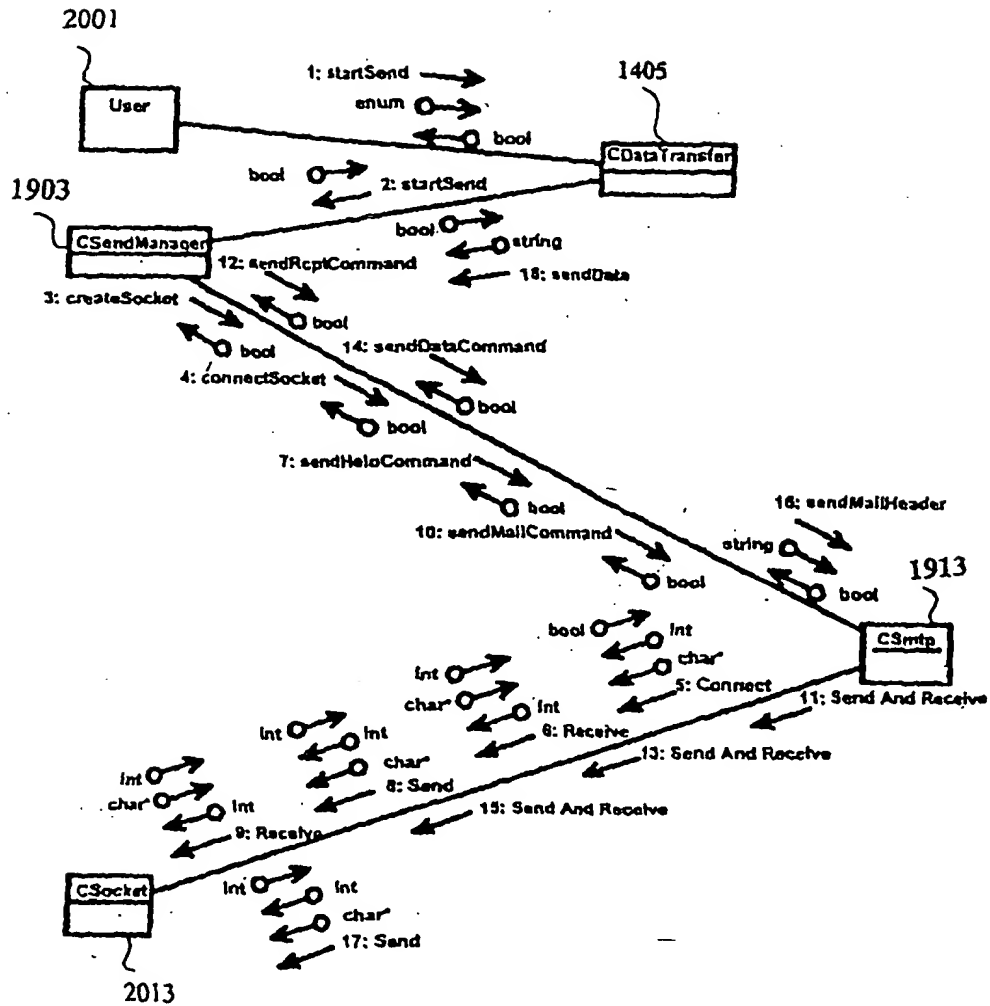


Figure 20 A

```
DataType, DeviceInformation
&&&&&
uniqueID, 00 00 aa 5e 1d fc
dateTime, 2000:08:14:14:57:15
Manufacturer, Xerox
Model, DocuPrint N4025 Network Laser Printer
...
PhoneNumber, 408-954-5445
&&&&&
uniqueID, 08 00 11 0f a5 33
dateTime, 2000:08:14:14:57:15
Manufacturer, Tektronix Inc
Model, Phaser 750P
...
PhoneNumber, 408-954-5429
#####
```

Figure 20B

```

DataType, StatusInformation
&&&&&
uniqueID, 00 00 aa 5e 1d fc
dateTime, 2000:08:14:14:57:15
errorCount, 0
lowPaper, 0
noPaper, 0
lowToner, 0
noToner, 0
doorOpen, 0
jammed, 0
offline, 0
serviceRequested, 0
configChangeCount, 14
pageLifeCount, 1025
&&&&&
uniqueID, 08 00 11 0f a5 33
dateTime, 2000:08:14:14:57:15
errorCount, 0
lowPaper, 0
noPaper, 0
lowToner, 0
noToner, 0
doorOpen, 0
jammed, 0
offline, 0
serviceRequested,
configChangeCount, 1
paperLifeCount, 202
#####

```

Figure 20C

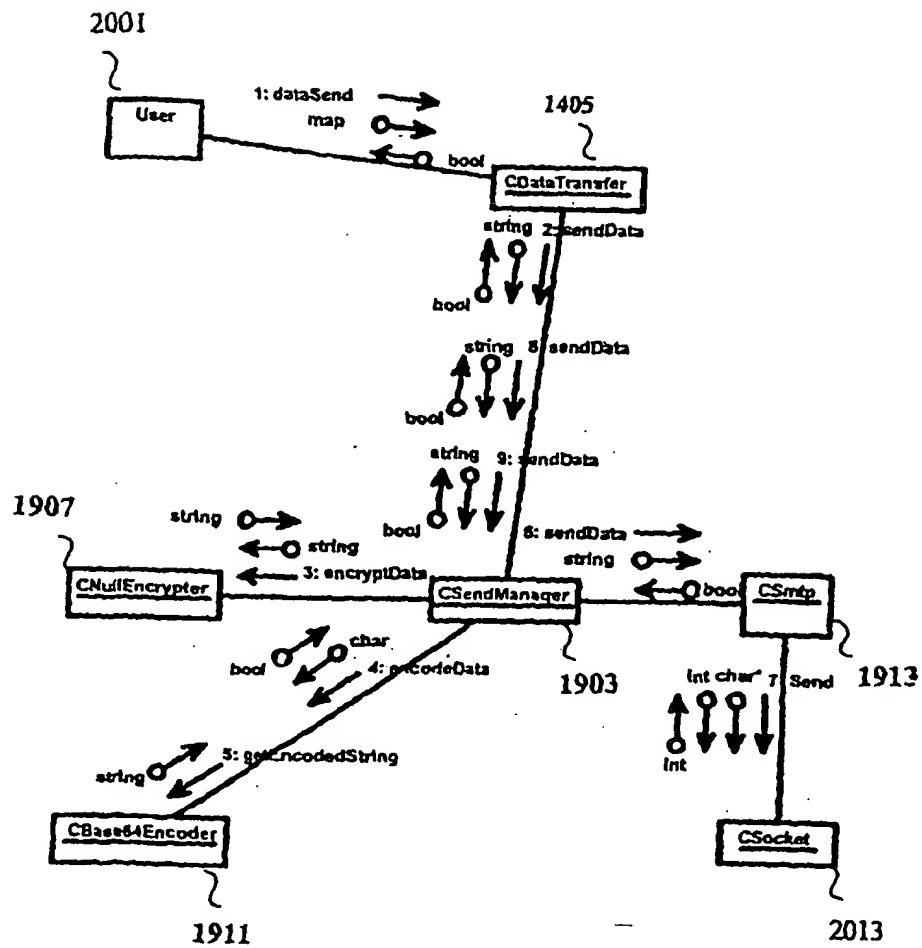


Figure 21

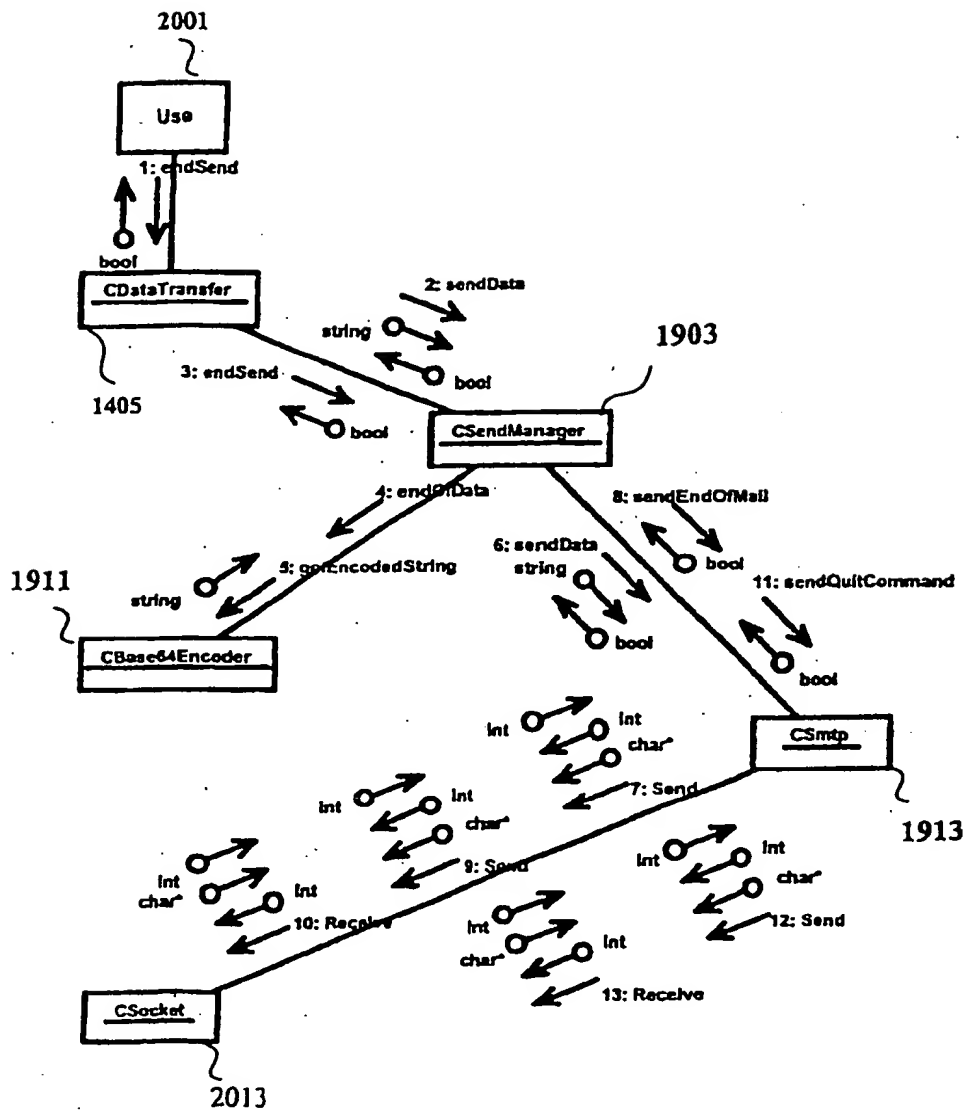


Figure 22

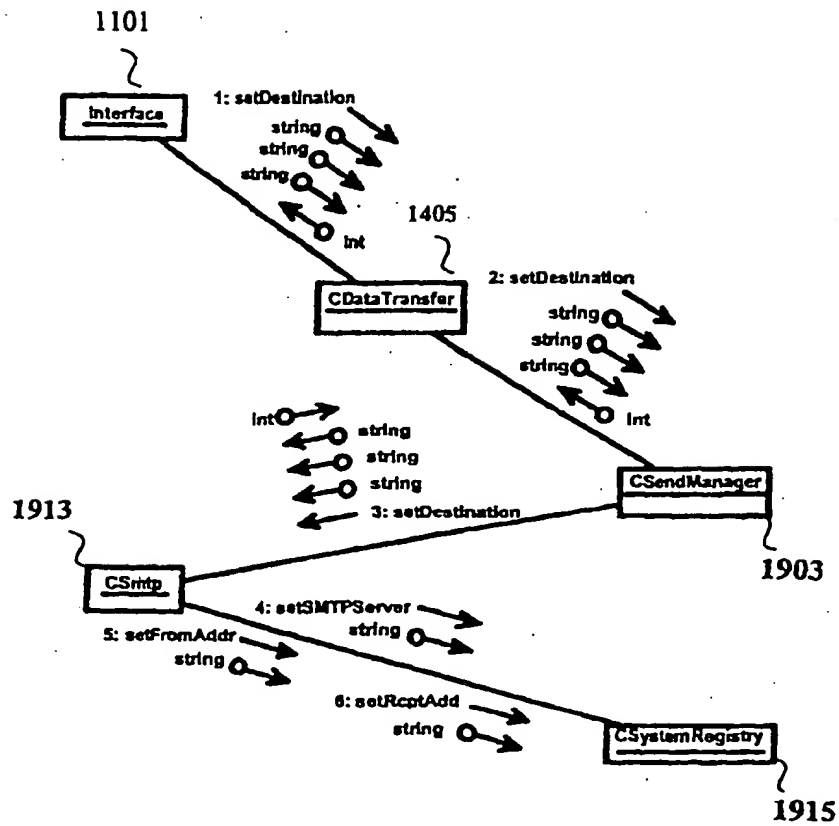


Figure 23

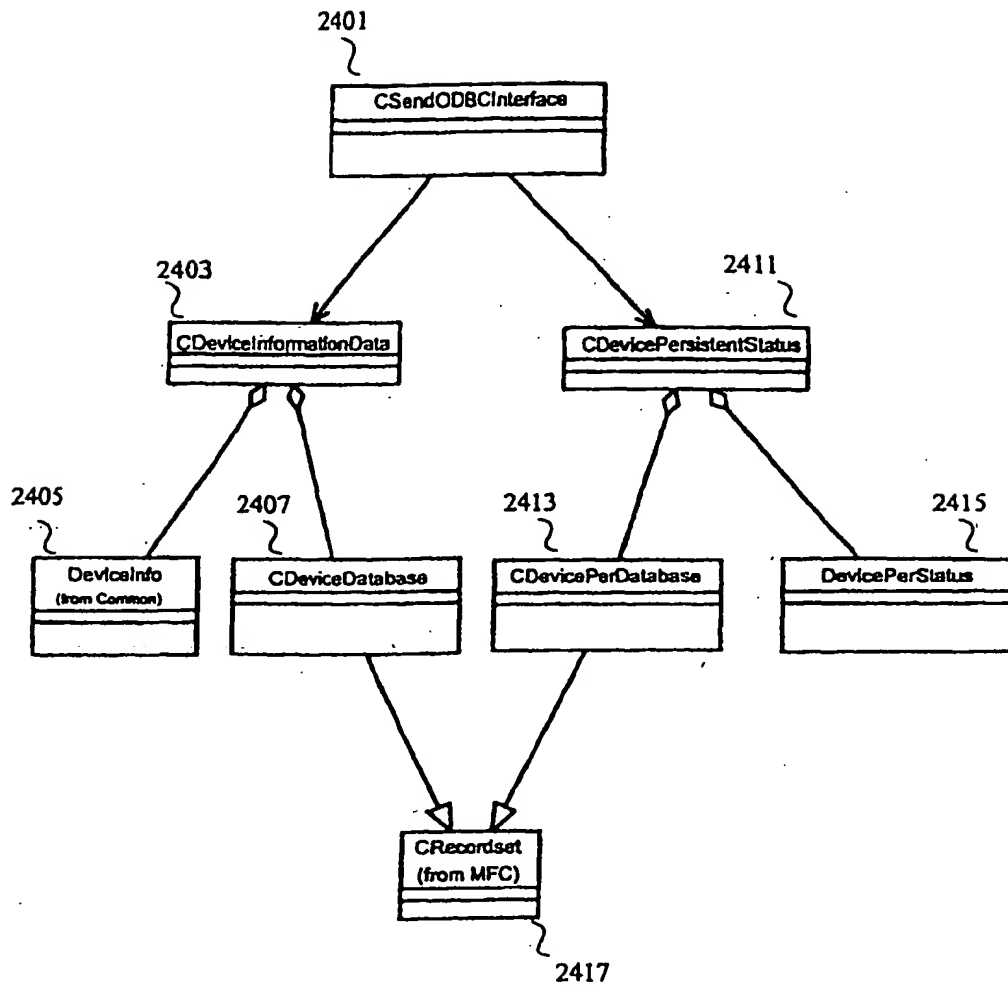


Figure 24



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 02 0531

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
Y A	EP 1 045 549 A (IBM) 18 October 2000 (2000-10-18) * abstract * * column 2, line 19 - column 4, line 18 * * column 5, line 18 - column 7, line 50 * ---	1-10, 18-27 11-17, 28-51	H04L12/24 H04L12/26 H04L12/58
Y	EP 0 918 411 A (HEWLETT PACKARD CO) 26 May 1999 (1999-05-26) * abstract * * column 2, line 3 - column 3, line 34 * * column 4, line 26 - column 5, line 52 * ---	1-10, 18-27	
A	WO 99 30467 A (ELDERSON HERMAN) 17 June 1999 (1999-06-17) * abstract * * page 2, line 24 - page 6, line 25 * ---	1-10, 18-27	
A	US 6 008 805 A (LAND ROBERT A ET AL) 28 December 1999 (1999-12-28) * abstract * * column 4, line 62 - column 6, line 44 * * column 15, line 54 - column 17, line 30 * ---	1-51	TECHNICAL FIELDS SEARCHED (Int.Cl.7) H04L
A	HEYWOOD P: "ADDED INSIGHT INTO CARRIER NETWORKS" DATA COMMUNICATIONS, MCGRAW HILL. NEW YORK, US, vol. 26, no. 5, 1 April 1997 (1997-04-01), pages 31-32, XP000659592 ISSN: 0363-6399 * the whole document * -----	1-51	
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 23 January 2003	Examiner Stergiou, C
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	

EPO FORM 1503 (01.02.98) (P04001)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 02 02 0531

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

23-01-2003

Patent document cited in search report		Publication date		Patent family member(s)	Publication date
EP 1045549	A	18-10-2000	EP	1045549 A1	18-10-2000
			JP	2000354035 A	19-12-2000

EP 0918411	A	26-05-1999	US	2002049693 A1	25-04-2002
			EP	0918411 A1	26-05-1999
			JP	11316725 A	16-11-1999

WO 9930467	A	17-06-1999	NL	1007709 C2	08-06-1999
			AU	1512199 A	28-06-1999
			WO	9930467 A1	17-06-1999

US 6008805	A	28-12-1999	AU	3796197 A	10-02-1998
			WO	9804067 A1	29-01-1998

EPO FORM P0455

For more details about this annex - see Official Journal of the European Patent Office, No. 12/82